Universidade de Aveiro
Departamento de Eletrónica, Telecomunicações e Informática,
2017

Bruno Filipe Dos Santos Faria

# Desenvolvimentos de uma nova abordagem em Inteligência Artificial para Deteção de Anomalias

# Developments of a new Artificial Intelligence approach for Anomaly Detection

Bruno Filipe Dos Santos Faria

# Desenvolvimentos de uma nova abordagem em Inteligência Artificial para Deteção de Anomalias

# Developments of a new Artificial Intelligence approach for Anomaly Detection

Tese apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Doutor em Engenharia Informática, realizada sob a orientação científica de Fernão Rodrigues Vístulo de Abreu, Professor Auxiliar do Departamento de Física da Universidade de Aveiro e co-orientação de André Ventura da Cruz Marnoto Zúquete, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio pela Bolsa de Doutoramento Ref. SFRH/BD/79865/2011

**o júri / the jury**

presidente / president

Doutor **João Filipe Colardelle da Luz Mano**
Professor Catedrático da Universidade de Aveiro (por delegação da Reitora da Universidade de Aveiro)

vogais / examiners committee

Doutor **Ernesto Jorge Fernandes Costa**
Professor Catedrático, Departamento de Engenharia Informática, Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Doutor **Paulo Alexandre Ribeiro Cortez**
Professor Associado com Agregação, Departamento de Sistemas e Informação, Escola de Engenharia da Universidade do Minho

Doutor **Manuel Eduardo Carvalho Duarte Correia**
Professor Auxiliar, Departamento de ciências e computadores, Faculdade de Ciências da Universidade do Porto

Doutor **Fernão Rodrigues Vístulo de Abreu**
Professor Auxiliar da Universidade de Aveiro **(orientador)**

Doutor **Paulo Jorge Salvador Serra Ferreira**
Professor Auxiliar, Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

**agradecimentos / acknowledgements**

Ao meu orientador Doutor Fernão Vístulo de Abreu pelo apoio incondicional e pelo enorme tempo despendido em discussões indispensáveis para o desenvolvimento deste trabalho.

Ao meu co-orientador Doutor André Zúquete pela grande ajuda e disponibilidade que sempre manifestou para discutir questões tanto relacionadas com a aplicabilidade do método como de âmbito mais geral.

Aos meus amigos pela compreensão, apoio e incentivo.

Aos restantes membros do júri por se terem disponibilizado a arguir esta tese.

**Resumo**

Este trabalho visou o desenvolvimento do modelo de frustração celular para aplicações à segurança informática. Neste âmbito foram desenvolvidos os processos necessários para materializar o modelo de frustração celular num algoritmo semi-supervisionado de deteção de anomalias. É por seguida efetuada uma comparação da capacidade de discriminação do algoritmo de frustração celular com algoritmos do estado de arte, nomeadamente máquinas de vetores de suporte e florestas aleatórias (com sigla em inglês de SVM e RF, respetivamente). Verifica-se que nos casos estudados o algoritmo de frustração celular obtém uma capacidade de discriminação de anomalias semelhante, senão melhor, que os algoritmos anteriormente descritos. São ainda descritas otimizações para reduzir o elevado custo computacional do algoritmo recorrendo a novos paradigmas de computação, i.e. pelo uso de placas gráficas, assim como otimizações que visam reduzir a complexidade do algoritmo. Em ambos os casos foi verificada uma redução do tempo computacional. Por fim, é ainda verificado que as melhorias introduzidas permitiram que a capacidade de discriminação do algoritmo se tornasse menos sensível à perturbação dos seus parâmetros.

**Abstract**

This work sought to develop the cellular frustration model for computer security applications. In this sense, the required processes to materialize the cellular frustration model in a semi-supervised anomaly detection algorithm were developed. The discrimination capability of the cellular frustration algorithm was then compared with the discrimination capability of state of the art algorithms, namely support vector machines and random forests (SVMs and RFs, respectively). In the studied cases it is observed that the cellular frustration algorithm exhibits comparable, if not better, anomaly detection capabilities. Optimizations to reduce the high computational cost that rely on new computational paradigms, i.e. by the use of graphic cards, as well as optimizations to reduce the algorithm complexity were also described. In both cases it was observed a reduction of the computational time required by the algorithm. Finally, it was verified that the introduced improvements allowed the anomaly detection capability of the algorithm to become less sensitive to the perturbation of its parameters.

# Contents

CHAPTER 1

# Introduction

The enormous proliferation of the internet and its increasing access simplicity has made internet one of the worlds biggest economic driving forces. Its use to shop, do business banking and communicate has shaped the way of how people manage their lives. Unfortunately, this proliferation creates opportunities for malicious individuals seeking to exploit the systems' security flaws and attack them for some benefit. Intrusion Detection Systems (IDSs) were created to stop the initiatives of such individuals. IDSs provide a clear indication about passed, in course or probable intrusions. To detect an intrusion, the majority of IDSs assume that the attack pattern is known, which is quite efficient for detecting known attacks but fails on new attack paradigms. Furthermore, since attackers are always trying to find new security flaws, cataloguing all types of intrusion patterns becomes impracticable on the long term.

Anomaly detection approaches present a clear advantage over attack pattern matching approaches. Instead of flagging an intrusion based on its known attack pattern, these type of approaches use the system "normal" behaviour to detect "abnormal" behaviour inherent to new attack paradigms. To achieve detection it is crucial to identify a characteristic that is stable under normal, legitimate behaviour, and is only perturbed by attacks [1]. However, it is unclear whether such a characteristic exists and since attackers are always searching for new forms of attack it is doubtful that this strategy isn't exploitable. In reality an anomaly may not be the result of the perturbation of a single characteristic, but rather its perturbation in a context, i.e how it changes in relation to others. It is not because an auto-mobile engine is operating over the "ideal" operating temperature (i.e. indicator over the middle of temperature gauge or approximately at 90 °C) that the engine is going to break. This may be the result of a decision made by the driver to go in a higher velocity than he would normally go. An anomalous behaviour would

be a higher temperature of the engine associated with slow movement. Current anomaly detection algorithms are not ideal to detect this type of situations. In order to build an anomaly detection model, they either use information about passed documented anomalies which don't necessarily exhibit the same behaviour of unseen anomalies, or make assumptions to what an anomaly is. One such example is one-class support vector machines which assume that "normal" data is concentrated and "abnormal" data are not [2].

In this work I try to address the problem of anomaly detection from a different perspective. Rather than using methods that make assumptions or use information of what an anomaly is I will use an immune inspired algorithm. The reason is that just like an IDS the task of the immune system is to discern between "normal" and "abnormal" using only "normal" information. The most popular artificial immune system (AIS) approach to intrusion detection was developed by Stephany Forrest *et al.* [3]. The method assumes that perfect discrimination of strange-known elements is unattainable and consequently builds the minimal detector repertoire, through negative selection, that can achieve a predefined detection failure rate of 5% [4, 5]. A strange element is identified when there is a match between its representation string and any of the detector strings. In general, thinking of practical applications in intrusion detection with this algorithm is difficult, since the number of detectors required increases exponentially with string length, but see [1, 6]. Furthermore, the method was developed for detecting strange elements and not perturbations on the context of known ones, which in itself can be defined as an anomaly [7].

A conceptually new approach, the Cellular Frustration Model (CFM), was proposed in [8, 9] by the main supervisor of this thesis. The CFM shares the same goals of Forrest algorithm while avoiding some of its problems, but departs from it in the sense that assumes a set of agents in interaction. What characterizes the interactions is that they have a short duration, and consequently the agents are considered to be in a high reactivity state. When a strange element appears or when the context of known ones changes the agents will start behaving differently, i.e. exhibiting higher interaction times, and as a result lowering the high reactivity state. It is this change in the reactivity that forms the basis for detection on the CFM.

The work develop here follows the work of two other students [10, 11]. These students have shown in their theses that it was possible to perfectly discriminate between known elements and strange elements (i.e. self-nonself discrimination). In [11] it was suggested that the CFM could do even more than just detecting strange elements, it was suggested that it could be used for anomaly detection. Mostardinha demonstrated this behaviour by analysing how the high reactivity state changed in the presence of different contexts of known agents. Here, I continue her work by exploring how the CFM can be formalized as an anomaly detection algorithm

with the aim of applying it to intrusion detection. This goal will be accomplished in several small steps, which are described below and are accomplished through the chapters of this thesis:

- To further develop the CFM and materialize it in anomaly detection algorithm. This involves searching for strategies that could optimize the Cellular Frustration Algorithm (CFA) in terms of efficiency and detection accuracy. In particular, special emphasis is given to the mapping of information, i.e. how should the information be mapped onto the CFA, and to the required modifications that could improve detection accuracy;

- To make a proper comparison with state of the art algorithms. This involves reviewing the literature for data-mining algorithms, with special emphasis on anomaly detection, and select the ones that can be used for assessing the anomaly detection performance of the CFA;

- To validate the CFA in the context of intrusion detection by comparing its anomaly detection performance with the performance of a state of the art algorithm in a public standard available dataset and under the same assumptions.

## 1.1 Thesis organization

This thesis is organized as follows. Chapters 2 to 4 are the introductory chapters and aim to describe the state of art of the several algorithms. Chapters 5 to 9 are dedicated to investigate the viability of CFM for anomaly detection. Finally, chapter 9 summarizes the main conclusions of this work and details possible future research directions.

I begin the introductory chapters by introducing the reader to the artificial immune inspired algorithms field (chapter 2). This chapter briefly describes the current accepted theoretical model for how the immune system performs pathogen detection, an algorithm developed from this view (i.e the Negative Selection Algorithm (NSA)), and the underline ideas of the algorithms exploring Polly Matzinger danger theory [12] and Grossman Tunnable Activation Threshold (TAT) hypothesis [13]. Then, in chapter 3 the two most widely known algorithms in the data-mining field for classification/anomaly detection, random forests (RF) and support vector machines (SVM), are described in detail. Special emphasis is given to these algorithms as they are extensively used through out this thesis for anomaly detection performance assessment. I end the introductory chapters by providing the necessary background in the CFA required to understand the developments made in this thesis, i.e. by

describing how the cellular frustration concept and model were materialized in an algorithm that performs self-nonself discrimination. The remaining chapters of this thesis are dedicated to the further development of the CFA and have been written as papers for international peer reviewed journals. All these papers have been reworked, so that they are presented in a consistent style and format in this thesis. One aspect that should be stressed is that because each chapter has been written as a paper, the bibliography for each chapter is at the end of the respective chapter.

Chapter 5 is focused on showing that the CFM can perform more than self-nonself discrimination it can also be used for detecting contextual changes. To accomplish this goal a modification - concerning how information is sensed by the CFM agents - is proposed and the different types of contextual perturbations analysed. This chapter constitutes the first attempt to describe the CFM as an anomaly detection algorithm and is part of a paper already submitted and expected to appear as:

> Bruno Filipe Faria, Patrícia Mostardinha, and Fernão Vístulo de Abreu. Can the Immune System Perform a t-Test? *PLOS ONE*, 12(1), jan 2017. doi: 10.1371/journal.pone.0169464

Chapter 6 is dedicated to the materialization of the CFM in an anomaly detection algorithm. To this purpose, some of the immunological constraints within the CFM are relaxed and the impact of the several algorithm parameters on the detection precision discussed. This chapter was submitted as:

> B. F. Faria and F. Vístulo Abreu. Cellular frustration algorithms for anomaly detection applications. *(submitted)*, 2016

Chapter 7 investigates the applicability of the CFA for intrusion detection. In this chapter the two most common strategies for program behaviour are analysed and implemented. The anomaly detection performance of the CFA is then assessed by comparison with the performance achieved by one-class SVMs on the MIT Lincoln 1999 DARPA dataset [16]. This chapter was submitted as:

> B. F. Faria, A. Zúquete, and A. M. Lindo. Intrusion detection using the cellular frustrated framework. *(submitted)*, 2016

Finally, chapter 8 is dedicated to explore the use of new computational paradigms, namely Graphical Processing Units (GPUs). In this chapter it is investigated if the CFA can be parallelized without compromising the dynamical characteristics of the algorithm and whether the parallelized algorithm implemented on a GPU can decrease the required computational time. Chapter 8 has been submitted as:

> B. F. Faria and F. Vístulo de Abreu. Computation of maximally frustrated populations with GPUs. *under revision*, 2016

## 1.2 Bibliography

[1] Stephanie Forrest, Steven Hofmeyr, and Anil Somayaji. The Evolution of System-Call Monitoring. In *2008 Annual Computer Security Applications Conference (ACSAC)*. Institute of Electrical & Electronics Engineers (IEEE), dec 2008.

[2] B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, pages 582–588, 2000.

[3] Stephanie Forrest, Alan S. Perelson, Lawrence Allen, and Rajesh Cherukuri. Self-Nonself Discrimination in a Computer. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, SP '94, pages 202–212, Washington, DC, USA, 1994. IEEE Computer Society.

[4] Zongxing Xie, Thiago Quirino, Mei-Ling Shyu, Shu-Ching Chen, and LiWu Chang. A Distributed Agent-Based Approach to Intrusion Detection Using the Lightweight PCC Anomaly Detection Classifier. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing -Vol 1 (SUTC'06) - Volume 01*, SUTC '06, pages 446–453, Washington, DC, USA, 2006. IEEE Computer Society.

[5] Patrik D'haeseleer, Stephanie Forrest, and Paul Helman. An Immunological Approach to Change Detection: Algorithms, Analysis and Implications. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, SP '96, pages 110–119, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 0-8186-7417-2.

[6] Jungwon Kim and Peter J. Bentley. An Evaluation of Negative Selection in an Artificial Immune System for Network Intrusion Detection. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO*, pages 1330–1337. Morgan Kaufmann, 2001.

[7] Thomas Stibor, Philipp Mohr, Jonathan Timmis, and Claudia Eckert. Is Negative Selection Appropriate for Anomaly Detection? In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, GECCO '05, pages 321–328, New York, NY, USA, 2005. ACM. ISBN 1-59593-010-8.

[8] F. Vístulo de Abreu, E. N. M. Nolte-'Hoen, C. R. Almeida, and D. M. Davis. Cellular Frustration: A New Conceptual Framework for Understanding Cell-mediated Immune Responses. In *Proceedings of the 5th International Conference*

*on Artificial Immune Systems*, ICARIS'06, pages 37–51, Berlin, Heidelberg, 2006. Springer-Verlag.

[9] F. Vístulo de Abreu and P. Mostardinha. Maximal frustration as an immunological principle. *Journal of The Royal Society Interface*, 6(32):321–334, 2009. ISSN 1742-5689.

[10] André Lindo. Nonself detection in immune inspired models. Master's thesis, University of Aveiro, 2010.

[11] Patrícia Mostardinha. *Developments of the Cellular Frustration Approach to Anomaly Detection*. PhD thesis, University of Aveiro, 2012.

[12] P Matzinger. Tolerance, Danger, and the Extended Family. *Annual Review of Immunology*, 12(1):991–1045, apr 1994.

[13] Zvi Grossman. Cellular Tolerance as a Dynamic State of the Adaptable Lymphocyte. *Immunological Reviews*, 133(1):45–73, 1993. ISSN 1600-065X.

[14] Bruno Filipe Faria, Patrícia Mostardinha, and Fernão Vístulo de Abreu. Can the Immune System Perform a t-Test? *PLOS ONE*, 12(1), jan 2017. doi: 10.1371/journal.pone.0169464.

[15] B. F. Faria and F. Vístulo Abreu. Cellular frustration algorithms for anomaly detection applications. *(submitted)*, 2016.

[16] MIT Lincoln Laboratory. 1999 Attack Database, 1999. URL `https://www.ll.mit.edu/ideval/docs/attackDB.html`.

[17] B. F. Faria, A. Zúquete, and A. M. Lindo. Intrusion detection using the cellular frustrated framework. *(submitted)*, 2016.

[18] B. F. Faria and F. Vístulo de Abreu. Computation of maximally frustrated populations with GPUs. *under revision*, 2016.

CHAPTER 2

# Immune inspired algorithms

The human immune system is a natural system that remarkably protects the human body from disease. During its protection of the human body from pathogens, the immune system exhibits several types of phenomena (such as: memory, learning and recognition) that can be used in problem solving. From a problem solving perspective it makes sense to model the different processes and structures of the immune system. Today, there are several algorithms that model the immune system processes and apply them for problem solving. In what concerns anomaly detection the most widespread and studied algorithms are [1]: the negative selection algorithm, dendritic cell algorithm, and tunable activation thresholds algorithms. These algorithms explore different mechanisms and theoretical views of the immune system. To explain them, I start by describing the currently accepted theoretical model of how the immune system detects foreign agents.

## 2.1 Immunological inspiration

To protect the body from damage the immune system relies on two main defence mechanisms (adaptive and innate) [2, 3]. The innate immune system provides the first layer of protection against pathogens and is characterized by its non specific immune response. Non-specific in the sense that it is not particular to a given pathogen agent, even-though it may have evolved, by natural selection, in that sense. Protection in this layer is achieved through anatomic barriers (skin, eye-browns, etc), secretions (saliva, tears, etc), phagocytic cells such as macrophages, neutrophils and natural killer cells [4]. These mechanisms allow the innate immune system to kill the vast majority of the micro-organisms. However, when a micro-organism deceives the non-specific immune system or is not promptly eliminated by it, a secondary immune response, denoted as the adaptive immune response, can be triggered. This

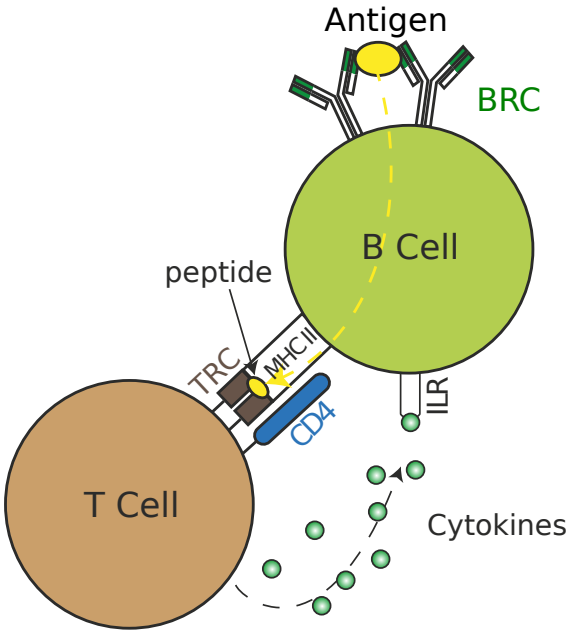


Figure 2.1: Representation of the surface elements that compose a T cell and an APC (B cell). Note the connection between the B cell receptor and the antigen and antigen presentation through MHC-II.

secondary response is often triggered through the stimulation of the adaptive system cells by the innate system cells.

The adaptive system cells have the ability to not only recognize a greater number of patterns but also to develop a much more specific response towards the intruder. To provide such response, the adaptive immune system relies on the lymphocytes (i.e. B and T Cells, antibody molecules and other molecules produced by lymphocytes). Both, B and T cells have surface receptor molecules that recognize antigen or more specifically, the antigen epitope. In the case of the B cells these receptors are denoted as antibodies (BCR) while in the T cells they are denoted as T receptors (TRC). The antigen recognition is dependent on the affinity between antigen and receptor and is processed according to the cell type. In B cells, antigen recognition is made when the antigen is found in its free (soluble) form in the blood stream or lymph. By contrast, T cells recognize the antigen in its processed form, presented by the antigen presenter cells (APCs) as a fragment.

To present the antigen fragment the APCs have major histocompatibility complex (MHC) molecules. These molecules are responsible for the collection and transportation of the peptide fragments from the APC intracellular environment to its surface. The peptide fragments result from the cells natural metabolism during its body function of collection and can have pathogenic source. For instance, dendritic cells and macrophages digest diverse materials (i.e. cells or remains of cells). Nonetheless, in most circumstances the peptide fragments come from the

body natural behaviour. Consequently, in order to avoid attacking the body own healthy cells the immune system must be able to discern the peptides accordingly to their source.

MHC molecules play a key role in the recognition process. They are responsible for the presentation of the peptide fragment to the T cells in the lymph nodes. The MHC molecules are divided in two classes. Class I molecules are specialized in the presentation of synthesized cell proteins, while class II molecules, which can only be found in APCs, are specialized in the presentation of molecule fragments collected in the extracellular environment. Class I molecules typically present the peptides to cytotoxic T cells (CTL - cytotoxic T lymphocytes), that will destroy the cell if they can bind to the complex formed by the peptide and the MHC class I molecule. On the other hand, class II molecules present the peptides to the T helper cells (Th). The activation of the Th cells, done through the binding with the MHC-peptide complex, will trigger the release of cytokines (proteins that control the intensity and duration of the immune response).

To control the immune response, cytokines apply a set of effects on the lymphocytes and other immune cells. For instance, antigen recognition added to the binding between a Th cell and a B cell triggers the appearance of a co-stimulation signal and consecutive release of cytokines (figure 2.1). Once, released the cytokines set off the division and differentiation of both B and Th cells in effector and memory cells. The B effector cells are denoted by B cell plasm and have the function of segregating large amounts of antibodies which allow the immune system to recognize cells as strange. An antigen-antibody complex, or a strange cell bond to antibodies, is promptly killed by phagocytic cells such as macrophages.

An immune response is characterized by very diverse phenomena, being the most extraordinary the exhibition of memory and the self-nonself discrimination. Memory exhibition is displayed whenever the immune system answers to a second intrusion by a same pathogen agent. Even if the first response to a pathogen agent takes several days, the second is almost immediate. In this case it is said that the immune system has acquired immunity to the pathogen.

Self-nonself discrimination is probably one of the most amazing challenges faced by the immune system. The immune system must be able to not only recognize the highest number possible of antigens, but at the same time avoid autoimmune responses. In order to avoid autoimmune responses (responses towards the body healthy cells) the immune system developed an education stage for the T lymphocytes. This education stage is applied to all T lymphocytes that migrate from the bone marrow to the thymus. Education of T lymphocytes happens in two stages. On a first stage T lymphocytes are positively selected based on their ability to bind to MHC molecules. The T lymphocytes that cannot bind to MHC molecules are

killed by apoptosis (programmed cell death). The cells that survive this stage face a second stage which is denoted as negative selection. Negative selection allows the immune system to avoid autoimmune responses (tolerance) by giving an apotosis signal to the cells that strongly bind with APCs presenting self peptides (peptides commonly found on the body). Consequently, only the cells that should form strong bonds with nonself peptides remain.

It is not apparent that T cells exiting the thymus will strongly bind with nonself peptides. This behaviour will be presented through the presentation of a computation model (the negative selection algorithm) inspired on the negative selection process. The negative selection algorithm has the merit of being one of the few quantitative models that describe and study the consequences of the negative selection process.

## 2.2 Negative Selection Algorithms

Negative selection was one of the first mechanisms to be explored and describes the vision on how the immune system performs T cell selection. Artificial negative selection is a computational description of this selection process and was first introduced by Stephanie Forrest and her collaborators [5]. Forrest introduced the negative selection algorithm for detecting viral sequences in computer systems. The idea was that by employing a set of detectors purporting the roles of T cells it would be possible to detect viral sequences. Within Forrest algorithm each detector is described by a fixed size binary sequence which will trigger activation if there is a match between the detector string and a presented string. To verify the matching a mathematical rule corresponding to the binding between an antigen and T cell is used.

The algorithm requires a censorship stage to build the detector set. During this stage randomly drawn detectors are compared with a self set. The self set is analogous to the self proteins that are stored on the thymus, in the sense that it contains the self elements used to test the detectors. All detectors that match self sequences are excluded from the detectors set. After the censoring stage the detector set can be used to monitor a given system (monitoring stage). Monitoring consists on the collection of new sequences from a pre-established system, translation into binary form and comparison with the detector set. Sequences matching any of the detectors in the detector set are considered nonself sequences and consequently call for an action to be taken.

Negative selection algorithms have however one main disadvantage, that is detection is not perfect. Perfect detection would require an exact correspondence between a detector sequence and the sequence being tested, which is not guaranteed by the matching rule. Guaranteeing an exact correspondence would require a

| |
|---|
| 0 1 1 **1 1 0 1** 0 |
| 0 1 0 **1 1 0 1** 1 |

Figure 2.2: Example of a matching when the number of bits to compare, $r$, is less or equal to 4. Note that if $r$ is bigger than 4 then not match occurs.

detector set with as much detectors as possible nonself sequences. In a system with 32 bits and 64 self sequences, $2^{32} - 64$ detectors would be required. Furthermore, the test of a sequence would require checking $2^{32} - 64$ detectors, which is impracticable and infeasible when a high number of sequences need to be tested. Instead, detection was made probabilist and as a result only $r$ bits need to match. Forrest denotes $r$ as the matching threshold and an example of a match when $r$ is smaller than 5 can be appreciated in Figure 2.2.

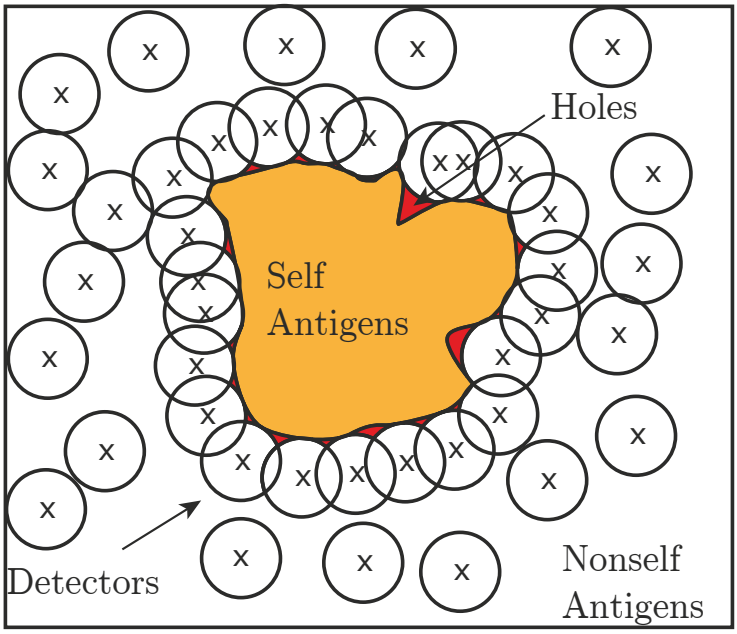


Figure 2.3: Illustration of how detectors surround the self region and cover the nonself one. Note the appearance of red regions that cannot be covered by any detector. The appearance of this regions is due to the elimination of candidate detectors matching self sequences in the censoring stage.

The probabilistic nature of detection makes negative selection algorithms vulnerable to classification mistakes – nonself sequences classified as self sequences. The regions of space originating the classification mistakes are denoted as holes (Figure 2.3) [6] which are the result of eliminating candidate detectors matching self sequences during the censoring stage. To better convey the appearance of holes consider an example where the $r$-contiguous bits rule is used with $r = 2$ and the

self set $S = \{110, 010\}$. The nonself sequence 011 will not be detect because any detector with 2 bits generated to detect it will also match a self sequence.

## 2.3 Dendritic cell algorithm

One algorithm that has been gaining widely recognition for anomaly detection in time series is the Dendritic Cell Algorithm (DCA) [7–9]. The DCA is an immune inspired algorithm developed around Polly Matzinger danger theory [10, 11]. The danger theory proposes a different view for how the immune system works. Rather than relying on the traditional self-nonself discrimination, it proposes that immune responses are triggered by danger or alarm signals which arise from damage or stress to the tissue cells. Damage is perceived as caused by invading micro-organisms, defects in host tissue or innate immune cells. Essentially, if pathogens do not cause damage they are tolerated. Regardless of the source, the theory assumes that the released signals are always the same, derive from the cell internal contents, and influence APCs behaviour. On signal presence, it culminates on lymph node naive T-cell activation and on absence it can lead to T-cell deletion or anergy.

Dendritic cells (DCs) are one suggested type of APCs capable of combining endogenous signals with foreign substances and respond accordingly. Dendritic cells exist in the body in different maturation states: mature, semi-mature and immature. The maturity state influences T-cell response. Immature DCs are responsible for collecting antigen in the tissue, where they are also subjected to endogenous signals. Depending on the combination of signals DCs evolve to either mature or semi-mature. Mature DCs trigger activation, while semi-mature DCs suppress activation. The resulting response is given by the combination of concentrations of semi vs mature DCs.

The DCA [7] builds upon this view by modelling antigen collection and dendritic cell maturation differentiation. Maturation differentiation in the DCA is accomplished by transforming the different input signal concentrations into concentrations of co-stimulatory molecules and cytokines (mature and semi-mature). Depending on the concentration difference of cytokines, DCs present the collected antigen, in the virtual lymph node, in a mature or semi-mature context. The antigen collection and DC virtual lymph node migration are cell dependent. In the virtual lymph node an antigen is flagged anomalous if it is presented more frequently in a mature context than in a semi-mature context.

Due to its nature applications of the DCA are, until the moment of this writing, mostly anomaly detection in time series. DCAs are of relatively straightforward implementation. However, one of the most difficult challenges one faces when implementing the DCA is the signal choice: What should be considered as a safe

signal? What should be considered as danger? Even though, much work has been done to answer this questions, they are still the subject of ongoing research.

## 2.4 Tunable Activation Thresholds

A rather different view that attempts to explain how the immune system works is the TAT model. TAT was proposed by Grossman *et al.* [12, 13] as an alternative to explain how lymphocytes adjust their responses to the micro-environmental context in which antigens are recognized. In Grossman model each cell has an activation threshold reflecting the cell stimuli history plus some critical value. Cells activate if the current stimuli exceeds the activation threshold, i.e. cell responsiveness is fully geared by the cells stimuli history. Cells continuously exposed to high external stimuli have high activation thresholds and consequently may get unresponsive. Furthermore, since there is an upper limit to the amount of stimuli a cell can receive, a sufficiently high activation threshold may prohibit cells from being activated. Cells are also not likely to be activated if the increase in stimuli is gradual, i.e. if the applied stimuli at each time instant is small.

According to Grossman, cell activation requires a strong stimuli that exceeds the activation threshold. Upon activation, only the cells that do not undergo differentiation – cells that become unresponsive, i.e. partly anergic and considered as memory cells – have the ability to regain responsiveness. To regain responsiveness for a subsequent challenge, the activation threshold of these cells must reach a sufficiently low value. However, if the infection agent persists the continued update of the activation threshold may result in no stimulus being able to exceed the threshold value.

Due to its intrinsic dynamic nature, cell behaviour driven by past and present contexts, the TAT model presents itself as a good candidate to perform anomaly detection in time series. Indeed, some authors have explored this route by developing an algorithm from the model [14–16]. The authors assumed a simple model where T cell activation was controlled by the activity of two enzymes, an excitation enzyme and a de-excitation enzyme, whose values reflect the recent temporal history of the signals received from APCs. A cell was activated if the value of the excitation enzyme superseded the de-excitation enzyme value. The authors applied this mechanism to temporal anomaly detection and concluded that even-though the algorithm could be used to detect temporal anomalies it had a major drawback. Anomalies need to be sporadic or the algorithm may fail to detect them. For anomaly-detection purposes this is a major drawback. Firstly, because anomalies should be detected independently of their frequency of appearance. Secondly, what constitutes an

anomaly is not know a priori, hence fine tuning the algorithm to detect some anomalies does not guarantee that the remaining types of anomalies can be detected.

Others tried to use Grossman ideas to build a classification algorithm for stationary data [17]. They used the same adaptable lymphocyte hypothesis as a data pre-processing step. The idea was to convert the dataset examples to profiles of T cell responses. Classification was then achieved by comparing the T cell response of test profiles with the response of training profiles. For comparison the authors relied on the k-nearest neighbour algorithm. They tested this idea on a series of datasets and reported that in most tests the developed algorithm could not perform better than a simple k-nearest neighbour.

## 2.5 Final remarks

In this chapter three different views of how the immune system performs pathogen detection have been briefly presented. The different views, either describe the immune system as able to perform self-nonself discrimination or self-abnormal self discrimination. However, and even-though the effort into improving these views, it is still unknown whether they can conciliate self-nonself discrimination with self-abnormal self discrimination.

The algorithms developed from these views have been successfully applied to a range of applications. However, probably due to the current flawed views of the immune system inner-workings, these algorithms suffer from a number of issues. Either perfect self-nonself discrimination is unattainable or what constitutes safe and danger signals is unknown or even unresponsiveness due to the continuously high applied stimuli. All these cases contribute to less than ideal algorithms compromising their use on an even wider range of applications. In this work it is shown that the cellular frustration model can conciliate both the self-nonself and the self-abnormal self mechanisms and consequently can compete with state of art data-mining algorithms.

## 2.6 Bibliography

[1] D. Dasgupta, Z. Ji, and F. Gonzlez. Artificial Immune System (AIS) Research in the Last Five Years. In *Proceedings of the International Conference on Evolutionary Computation*, December 2003.

[2] Dipankar Dasgupta and Fernando Nino. *Immunological Computation: Theory and Applications*. Auerbach Publications, Boston, MA, USA, 1 edition, 2008.

[3] J. Timmis, A. Hone, T. Stibor, and E. Clark. Theoretical advances in artificial immune systems. *Theorertical Computer Science*, 403(1):11–32, August 2008. ISSN 0304-3975.

[4] A.K. Abbas and A.H. Lichtman. *Basic Immunology: Functions and Disorders of the Immune System*. Elsevier/Saunders, Philadelphia, PA, 2010. ISBN 9781416055693.

[5] Stephanie Forrest, Alan S. Perelson, Lawrence Allen, and Rajesh Cherukuri. Self-Nonself Discrimination in a Computer. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, SP '94, pages 202–212, Washington, DC, USA, 1994. IEEE Computer Society.

[6] Patrik D'haeseleer, Stephanie Forrest, and Paul Helman. An Immunological Approach to Change Detection: Algorithms, Analysis and Implications. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, SP '96, pages 110–119, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 0-8186-7417-2.

[7] Julie Greensmith, Uwe Aickelin, and Steve Cayzer. *Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection*, pages 153–167. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-31875-0.

[8] Julie Greensmith and Uwe Aickelin. The deterministic dendritic cell algorithm. In *International Conference on Artificial Immune Systems*, pages 291–302. Springer, 2008.

[9] Julie Greensmith and Uwe Aickelin. Artificial dendritic cells: multi-faceted perspectives. In *Human-Centric Information Processing Through Granular Modelling*, pages 375–395. Springer, 2009.

[10] P Matzinger. Tolerance, Danger, and the Extended Family. *Annual Review of Immunology*, 12(1):991–1045, apr 1994.

[11] Polly Matzinger. The danger model: a renewed sense of self. *Science*, 296 (5566):301–305, 2002.

[12] Zvi GROSSMAN and WILLIAM E. PAUL. Adaptive cellular interactions in the immune system: The tunable activation threshold and the significance of subthreshold responses. *Proceedings of the National Academy of Sciences of the United States of America*, 1992.

[13] Zvi Grossman. Cellular Tolerance as a Dynamic State of the Adaptable Lymphocyte. *Immunological Reviews*, 133(1):45–73, 1993. ISSN 1600-065X.

[14] Mário Antunes and Manuel Correia. *TAT-NIDS: An Immune-Based Anomaly Detection Architecture for Network Intrusion Detection*, pages 60–67. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-85861-4.

[15] Mário J. Antunes and Manuel E. Correia. Temporal Anomaly Detection: An Artificial Immune Approach Based on T Cell Activation, Clonal Size Regulation and Homeostasis. In *Advances in Experimental Medicine and Biology*, pages 291–298. Springer Science + Business Media, 2010.

[16] Mário J. Antunes and Manuel E. Correia. *Self Tolerance by Tuning T-Cell Activation: An Artificial Immune System for Anomaly Detection*, pages 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-32615-8.

[17] Paul S. Andrews and Jon Timmis. *Tunable Detectors for Artificial Immune Systems: From Model to Algorithm*, pages 103–127. Springer New York, New York, NY, 2010. ISBN 978-1-4419-0540-6.

CHAPTER 3

# Data Mining Algorithms

Data-mining is a step in the broader field of knowledge discovery from data (KDD). It describes the (semi) automatic process of digging through large amounts of data to find patterns [1]. In practice, pattern finding in data mining serves either of two main goals, description or prediction. Description focuses on finding interpretable patterns describing the data. For instance, what describes an earthquake is the shaking and vibration at the surface of the earth resulting from underground movement along a fault plane or from volcanic activity. Prediction, on the other hand, involves using the data knowledge, current or past, to predict some unknown quantity or state. In weather forecast, this implies collecting quantitative data about the current state of the atmosphere, such as: temperature, wind, pressure and humidity, with the intent of predicting the future state of the weather. A variety of particular data mining methods can be used to achieve either of the goals. In this work I focus mainly on description algorithms, such as classification and anomaly detection algorithms. For this purpose, I describe, in the next sections, two of the most widely known methods: random forests and support vector machines.

## 3.1 Random Forests

The random forests algorithm was introduced by Leo Breiman and Adele Cutler to essentially perform classification and regression tasks [2, 3]. The algorithm operates in two stages that I denote as training and evaluation. In training the algorithm combines bootstrap aggregating with random selection of features to grow an ensemble of decision trees. These decision trees are then used in evaluation to either output the class which is the mode of the classes (classification) or mean prediction (regression) of the individual trees. The fundamental building block of the random forests algorithm is decision trees. As a result I will first describe the

Table 3.1: Dataset illustrating a scenario where a person is allowed access into a building based on eye and hair colour.

| Hair colour | Eye colour | Has access? |
|---|---|---|
| black | brown | Yes |
| blond | brown | Yes |
| blond | blue | Yes |
| blond | green | No |
| black | blue | No |
| brown | brown | Yes |

decision trees algorithm and only then the modifications introduced by Breiman that lead to the random forests algorithm.

## Decision trees

The term decision trees is a broad term whose definition is entirely dependent on the field. In data mining, a decision tree is a description and predictive model which can be used to represent both classification and regression models. On the other hand, in fields such as operations research, it is a hierarchical model of decisions and consequences which is intended to aid a decision maker identify the most likely strategy to reach a specific goal. In this work I concentrate only on classification and regression trees (CART) as these were the ones developed and used in random forests by Leo Breiman *et al.* [4].

The denomination classification or regression tree depends entirely on the task. If the task is classification then the decision tree is denoted as classification tree. If on the other hand is regression, then the decision tree is denoted as regression tree. Regardless of the notation, both classification and regression trees operate in a similar way. That is, they use a series of yes or no questions to determine the class or mean prediction of a sample. More specifically, they are a flowchart-like structure (tree) where each node represents a test on an attribute (e.g. whether the temperature is greater than 30 °C), each branch represents the test outcome and the leaf node represents a class label (e.g. hot or cold). To build this flowchart like structure a training stage is required. In the training stage, the algorithm searches for the combination of feature/value that best divides the training samples in two groups. That is, the value that minimizes category heterogeneity in the two groups. This process is then repeated for each group until a stopping criteria is met. To easily convey how this process is accomplish consider the dataset on Table 3.1.

This dataset describes a scenario where access into a building depends on a

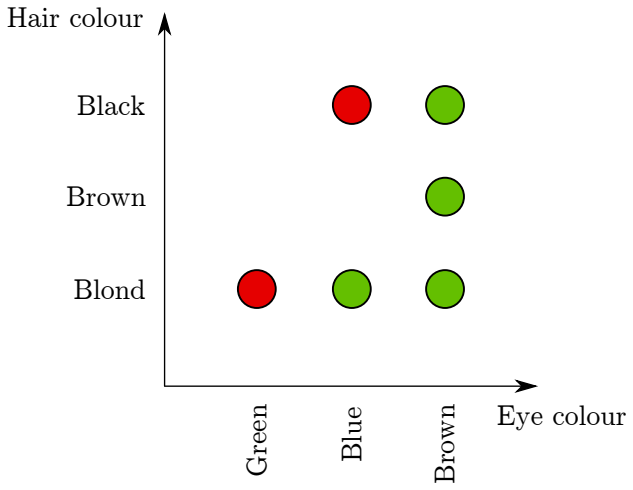


Figure 3.1: Bi-dimensional representation of the dataset illustrated on Table 3.1. The green circles represent persons with access to the building while the red circles represent persons without access.

persons eye and hair colour. Hence, a classification tree should be used. It can be observed that the dataset does not describe all possible scenarios. For instance, what action should be taken when a person with green eyes and black hair appears? The reader will observe that the classification tree will also cover this case. However, first it needs to be generated. The tree generation process is more easily conveyed through a representation of the above dataset in a 2 dimensional plot (Figure 3.1).

From inspection of Figure 3.1 one can observe that any person with brown eyes is granted access into the building. Hence, the first combination of feature/value to be used is eye colour/brown. This is represented by the tree first node (1) on Figure 3.2-b) and by the line (1) on a). Node (1) has two child nodes. One child node that grants access to any person with brown eyes - a leaf node - and one for people with other eye colours - node (2). In node (2) there are three types of persons, two that don't have access and one that has. Separation on this node can be made by eye or hair colour as it will result in the same 2 versus 1 category division. Here, I chose to separate by hair colour. That is, if the person doesn't have brown eyes and has black hair then it should be denied access. This implies two child nodes, a leaf node denying access for people with black hair and another node for people that don't have black hair, node (3). In node (3), the last node, the division is straightforward as there are only two types of blond persons. One person with green eyes which does not have access and one with blue eyes with access. Hence, the division should be done on eye colour.

Notice that even-though the dataset in Table 3.1 did not contain all possible scenarios, the resulting tree has been generalized to handle them. For instance, a person with blue eyes and brown hair has access to the building. This access is the

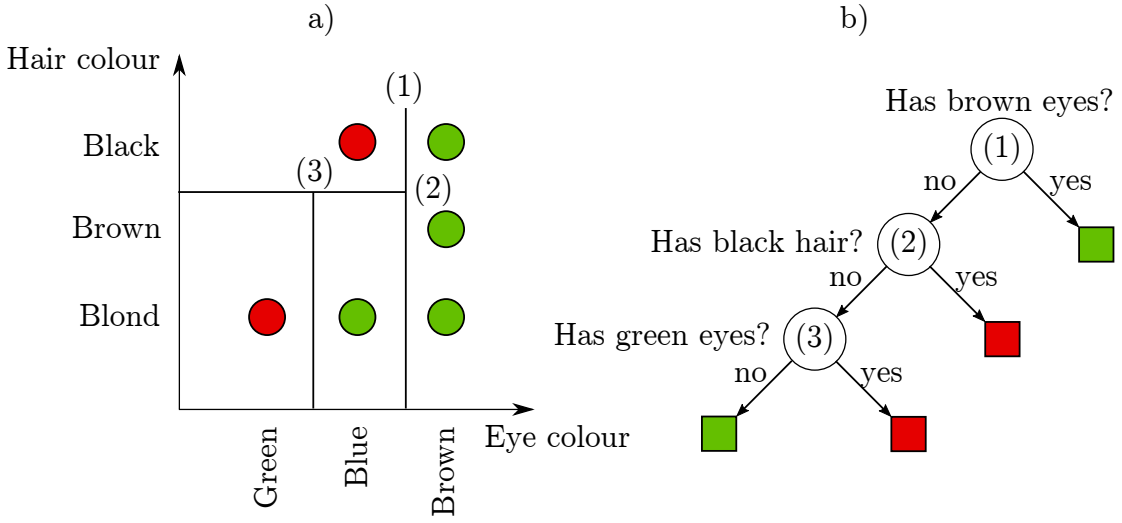


Figure 3.2: In a) the bi-dimensional representation of the dataset illustrated on Table 3.1 with sequence of questions required to allow/deny access into the building. The green circles represent persons with access to the building while the red circles represent persons without access. In b) the tree representation of the sequence of questions to be made to allow or deny access into the building.

result of the feature chosen to split on node (2). If instead of hair colour the eye colour was used then this person would not have access.

In the example above a greedy metric was used for choosing the combination of feature/value for splitting. However, in order for the algorithm to be of practical use the splitting metric needs to be mathematically formalized. In the literature several metrics for splitting are available [4, 5]: gini impurity, information gain, variance reduction, entropy, misclassification, amongst others. Nonetheless, here I will only explain gini impurity and variance reduction as these are the ones employed in CART.

The use of gini impurity or variance reduction is dependent on the task. For classification tasks a decision tree is constructed using the gini impurity metric, and for regression tasks where the category variable is continuous the variance reduction metric is used. Since, this thesis is mostly dedicated to classification and the above example was a classification example, I will start by describing the gini impurity metric and how it can be used in tree construction.

The gini impurity metric is the expected error of predicting the node class by randomly selecting the class label from the node class distribution. It can be viewed as a metric for node heterogeneity. A value of zero indicates that the samples on the node are all from the same class. In the same sense, a gini impurity of 0.5 indicates that there is the same number of samples from each of the individual classes. Mathematically the gini impurity is represented as:

Table 3.2: Dataset illustrating the decrease in gini impurity for each combination of feature/value in each node of the decision tree illustrated on Figure 3.2

| | Eye colour | | | Hair colour | | |
|---|---|---|---|---|---|---|
| Node | Blue | Brown | Green | Black | Blond | Brown |
| (1) | 0.0833 | 0.2778 | 0.2333 | 0.0833 | 0.0952 | 0.1000 |
| (2) | 0.1067 | - | 0.1067 | 0.1067 | 0.1067 | - |
| (3) | 0.5000 | - | 0.5000 | - | 0.1067 | - |

$$i(N) = \sum_{j}^{n_c} \sum_{k \neq j}^{n_c} P(w_j)P(w_k) \tag{3.1}$$

where $P(w_j)$ is the probability of selecting a sample from a class $j$, $P(w_k)$ is the probability of making a wrong selection and $n_c$ is the number of classes. The gini impurity is often simplified to:

$$i(N) = 1 - \sum_{j}^{n_c} P(w_j)^2. \tag{3.2}$$

One might question how the gini impurity can be used for selecting the best combination of feature/value for splitting. The question, is what is defined as the "best" combination for splitting? If it is considered that is the one that most reduces node class heterogeneity, then it is the combination of feature/value that most reduces gini impurity. Mathematically, this can be expressed as:

$$\Delta i(N) = i(N) - P_L i(N_L) - P_R i(N_R) \tag{3.3}$$

where $\Delta i(N)$ represents the change in node $N$ gini impurity, $i(N)$ the node gini impurity, $i(N_L)$ and $i(N_R)$ the left and right child nodes gini impurities and $P_L$ and $P_R$ the probability of having samples in the left and right child nodes, respectively. The aim in each node is to maximize $\Delta i(N)$.

In Table 3.2 I represent the $\Delta i(N)$ for each node of the classification tree illustrated on Figure 3.2 for each combination of feature/value. As it can be seen the maximum values of $\Delta i(N)$ are attained for the combination of features/values selected for the classification tree on Figure 3.2.

Regression trees do not have classes. Instead, there is a response variable assigned to each sample. This implies that metrics such as the gini impurity cannot be used. Splitting in regression trees is accomplished by minimizing the expected sum variance for two resulting nodes. Mathematically this is expressed as:

$$\Delta V(N) = V(N) - P_L V(N_L) - P_R V(N_R) \tag{3.4}$$

where $\Delta V(N)$ is the change in variance, $V(N)$ is the variance at node $N$, $P_L$ and $P_R$ the probability of having samples in the left and right child nodes and $V(N_L)$ and $V(N_R)$ the variance at the left and right child nodes. The aim is to maximize $\Delta V(N)$, because maximization of $\Delta V(N)$ leads to node variance minimization.

The last point to discuss on decision trees is when to stop growing the decision tree. If the tree is fully grown to the point where in each node there is exactly one training sample, then the decision tree is considered to be over-fitting to the training data. The problem is that over-fitting trees are generally not good predictors for samples that are not in the training set (i.e. they have poor generalization capability) [6]. Therefore, the decision tree needs to be optimized before being used for classification or regression. In the literature, there exist essentially two methods for optimizing decision trees, early stopping and pruning. Early stopping is the simplest of both methods as one just needs to stop growing the tree branch when the change in impurity (Equation (3.3)) or some other quantity of the node drops below a $\beta$ quantity. Pruning, on the other hand, grows the tree to its full extent and then it eliminates the tree branches that conduct to low generalization capability. To compute the tree generalization capability also referred to tree *true predictive power*, cross-validation is used during the tree growing process to separate the training set in two sets, the decision tree construction set, and testing set. Even-though there are several strategies in the literature for pruning I will not explain them as pruning is not required in random forests [2].

## From decision trees to random forests

Decision trees have several advantages. They require little data preparation, can handle numerical and categorical data and are simple to understand and interpret. However, despite all of these advantages they have three main limitations. Firstly, to achieve a high classification/regression performance it is required parameter tuning which is specific to the dataset. Secondly, even thought decision trees have low bias they have high variance, that is, a small perturbation on the training set can lead to a very different decision tree (over-fitting). Thirdly, the monothetic splitting implies that correlations between features may not be captured hence, the performance can be low. To overcome these limitations, Leo Breiman and Adele Cutler introduced random forests [2]. Random forests use decision trees as the base algorithm but introduce two modifications, bootstrap aggregating (bagging) and random selection of features which are explained below.

Instead of considering a single decision tree, random forests consider the result of an ensemble of decision trees to classify or predict the value of a sample. If the task is classification the resulting class is the class most voted by the several decision trees. If, on the other hand, is regression then it is the mean of the responses given by the different trees. To build the ensemble of decision trees bootstrap aggregating is used (bagging). Bagging is the action of building training sets by sampling with replacement from the original training set. More concretely, given a standard training set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^d$ of size $n$, bagging generates $m$ new training sets $X^i$ by sampling uniformly and with replacement from $X$. Each training set $X^i$ is then used to grow each decision tree on the ensemble (Algorithm 3.1). The idea is that combining the results of decision trees built on different parts of the training set reduces the ensemble variance [2]. That is, even-though the ensemble decision trees are different the combination of the different decision trees gives a smooth decision boundary that doesn't change much if the training set is perturbed [2].

The last modification introduced in random forests is random feature selection. Random feature selection was motivated by the work of Amit and Geman [7] and aims to decorrelate the decision trees grown from the different bagged training sets. The idea is that instead of searching for the feature that best splits the dataset over the feature space, one should only focus in a random subset of the feature space [2]. The reason is because if there is at least one feature that is strong predictor for the target class or response variable, then this feature will be selected in many of the grown decision trees causing them to be correlated. By randomly selecting $k$ features, in each node, to be optimized the probability of choosing the feature that is a strong predictor for the response variable (or target class) decreases. For classification problems $k$ is typically defined as the square root of the feature space, as for regression the typical value is one third of the feature space size [3]. Regardless, of the values suggested this is a free parameter that can range from one to the feature space size and is problem dependent. As such, it should be considered as a tuning parameter.

### Random forest algorithm pseudo-code

A standard implementation of the random forest algorithm for the classification task is illustrated in Algorithms 3.1, 3.2 and 3.3. Here, I only illustrate the pseudo-code for a classification task however, with minimal changes regression can also be covered. Note that, error handling, out of bag estimation and other features have not been included in the algorithm. The reason is that this is an illustration of the random forest algorithm which provides a starting point for a more complete and thorough algorithm.

**Algorithm 3.1** Main algorithm for a classification task, *train* should be set to *true* or $false$ depending on the task, i.e growing trees (training) or sample evaluating (classification), respectively.

**Require:** $X$ (training set), $Y$ (training set labels), $number\_trees$ ($m$)
**Require:** $tree^{ensemble}$, $T$, $number\_trees$ for classification (evaluation)
**function** RF($train, tree^{ensemble}, X, Y, number\_trees$)
  **if** *train* is *true* **then**
    **for** $i$ in 1 to $number\_trees$ **do**
      Draw $X^i$ from $X$ by sampling uniformly and with replacement
      Get the correspondent $Y^i$ for $X^i$
      $I \leftarrow 1$ to $length(Y^i)$
      $k \leftarrow \sqrt{length\ of\ X\ feature\ space}$
      $tree^{ensemble}[i] \leftarrow$ GENTREE($\emptyset, X^i, Y^i, I, 1, 0.5, k$)
    **end for**
    **return** $tree^{ensemble}$
  **else**
    **for** $i$ in 1 to number of samples in $T$ **do**
      **for** $j$ in 1 to $number\_trees$ **do**
        $temp\_class[j] \leftarrow$ GETTREECLASS($tree^{ensemble}[j], 1, X_i$)
      **end for**
      $class[i] \leftarrow$ most voted class in $temp\_class$
    **end for**
    **return** $class$
  **end if**
**end function**

The illustrated random forest algorithm is executed in two stages: a stage to grow the tree ensemble and a stage to evaluate samples. The first stage to be executed is training. Hence, the variable *train* should be set to true and the dataset $X$ with the correspondent labels $Y$ supplied. In this stage the number of trees to form the ensemble should also be given. The *gini threshold* is a global variable that is problem dependent. However, it is typically set to 0.1. The algorithm starts by checking if it should generate a tree ensemble. If yes, then the algorithm generates a bootstrap set of samples $X^i$ from $X$ and assigns the corresponding labels $Y$ to $Y^i$. Then, the algorithm sets the indexes of the samples to use in the tree generation process. Since the algorithm starts from the root node, this implies all sample indexes. The last value to be set before generating the tree is the number of features to randomly draw at the node splitting. Finally, the tree generation process is executed and the whole process repeated number of tree times.

The tree generation process (Algorithm 3.2) is a recursive algorithm that is centred in the node splitting process. It starts by checking if the gini impurity is below the gini threshold. If it is, the node is declared as as a leaf node and the leaf label is set to the most frequent label in the node samples. If not, then the algorithm

**Algorithm 3.2** Tree generation function pseudo code for a classification tree.

**Require:** *gini_threshold* can be globally set for all trees

**function** GENTREE($tree, X^i, Y^i, I, n, gi_n, k$)
  $tree.node \leftarrow n$
  **if** $gi_n \leq gini\_threshold$ **then**
    $tree.label[n] \leftarrow$ most frequent label in $Y^i[I]$
    $tree.leaf[n] \leftarrow$ yes
  **else**
    $tree.leaf[n] \leftarrow$ no
    $F \leftarrow$ sample $k$ features from $X^i$ feature space
    $\{F^{split}, v^{split}, gi_{decrease}, gi_l^{split}, gi_r^{split}\} \leftarrow \{0\}$
    $I_l^{split} \leftarrow \emptyset$, $I_r^{split} \leftarrow \emptyset$
    **for all** $f \in F$ **do**
      **for all** $v \in X_{If}^i$ **do**
        $P_l \leftarrow \sum_{j \in I}(X_{jf}^i < v)/length(I)$
        $P_r \leftarrow \sum_{j \in I}(X_{jf}^i \geq v)/length(I)$
        $I_l \leftarrow$ indexes of $I$ where $X_{If}^i < v$
        $I_r \leftarrow$ indexes of $I$ where $X_{If}^i \geq v$
        $gi_l \leftarrow$ gini impurity (Equation (3.1)) computed from $Y^i[I_l]$
        $gi_r \leftarrow$ gini impurity (Equation (3.1)) computed from $Y^i[I_r]$
        $\Delta gi = gi_n - P_l gi_l - P_r gi_r$
        **if** $\Delta gi > gi_{decrease}$ **then**
          $I_l^{split} \leftarrow I_l$
          $I_r^{split} \leftarrow I_r$
          $F^{split} \leftarrow f$
          $v^{split} \leftarrow v$
          $gi_{decrease} \leftarrow \Delta gi$
          $gi_l^{split} \leftarrow gi_l$
          $gi_r^{split} \leftarrow gi_r$
        **end if**
      **end for**
    **end for**
    $tree.F^{split}[n] \leftarrow F^{split}$
    $tree.v^{split}[n] \leftarrow v^{split}$
    $tree.child_l[n] \leftarrow n + 1$
    $tree \leftarrow$ GENTREE($tree, X^i, Y^i, I_l^{split}, n+1, gi_l^{split}, k$)
    $tree.child_r[n] \leftarrow tree.node + 1$
    $tree \leftarrow$ GENTREE($tree, X^i, Y^i, I_r^{split}, tree.node + 1, gi_r^{split}, k$)
  **end if**
  **return** *tree*
**end function**

randomly samples $k$ features and chooses the combination of feature/value that most decreases gini impurity (Equation (3.3)). For the value that most decreases gini impurity the algorithm registers the indexes of the samples that are below (left

child node) and above (right child node) the split value, that is, $I_l^{split}$ and $I_r^{split}$ respectively. The algorithm then recursively calls itself to divide the data from the left ($I_l^{split}$) and right ($I_r^{split}$) child nodes. Finally, when all the nodes have been transverse the algorithm returns the generated tree.

**Algorithm 3.3** Sample evaluation function.

```
1: function GETTREECLASS(tree, n, sample)
2:     if tree.leaf[n] is true then
3:         return tree.label[n]
4:     else
5:         if sample[tree.F^split[n]] < tree.v^split[n] then
6:             return GETTREECLASS(tree, [tree.child_l[n], sample)
7:         else
8:             return GETTREECLASS(tree, [tree.child_r[n], sample)
9:         end if
10:    end if
11: end function
```

After generating the tree ensemble, the algorithm is ready for evaluation. To evaluate a set of samples $X$, the *train* variable should be set to false and the tree ensemble should be given. Then for each sample in $X$ the algorithm evaluates the class given by each tree in the ensemble. For a given tree the class is obtained by recursively evaluating the function *GETTREECLASS* on the sample (Algorithm 3.3). Starting from the root node, the *GETTREECLASS* function evaluates if the sample value of the node splitting feature is below or above the node define splitting value. If it is below the function recursively calls itself on the tree left child node. If it is equal or above then it recursively calls itself on the tree right child node. The function terminates when the tree leaf node is reached by returning the leaf class label as the sample belonging class. After all trees have been evaluated the sample class is assigned to the most frequent class given by the tree ensemble.

## 3.2 Support Vector Machines

Support vector machines (SVMs) are a data mining tool introduced by Vapnik *et al.* [3] to perform classification, regression and other tasks. Just like random forests, SVMs typically require two stages: training and evaluation. Training is used to find the model parameters that allow the algorithm, in evaluation, to classify or predict the value of a sample. These two stages will be explained in more detail in the next subsections. However, contrarily to the last section here I only explain classification. The reason is that contrarily to random forests, regression in SVMs needs to be explained separately, which introduces additional complexity and falls out of the

scope of this thesis. I start this chapter by describing the linear separable case of support vector machines.

## The linear separable case

Support vector machines (SVMs) were introduced by Vapnik *et al.* [8, 9] as a method to perform binary classification. The underline idea in SVMs consists in finding a separating hyperplane that maximizes the separating margin between both categories. The feature vectors that lie on the defined maximum separating margin are denoted as support vectors. Hence, classifiers that exploit this property are designated as support vector machines.

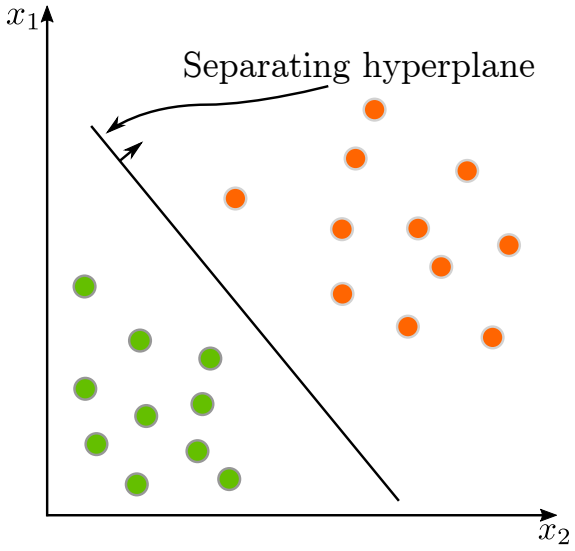


Figure 3.3: Illustration of a linearly separable binary classification problem. In this illustration, the dots represent the samples, the colours the two different classes and the line represents the desired hyperplane to separate both classes. Note that samples are represented in a $\mathbb{R}^2$ space however, representing the samples class $y_i$ requires a $\mathbb{R}^3$ space, where the separating hyperplane resides. The illustrated separating line is the hyperplane sliced at $y = 0$.

To understand how the role of the hyperplane and why it needs to be defined with maximum margin an example will be used. Consider then, the set of training samples illustrated in Figure 3.3. The dataset illustrates a binary classification problem where the aim is to correctly identify the samples class (illustrated with colours on Figure 3.3). Support vector machines consider an hyperplane to achieve this categorization. That is, if each sample $i$ is represented by a set of features $\mathbf{x}_i \in \mathbb{R}^2$ and as an assigned label (class) $y_i = \pm 1$, then SVMs aim to find the hyperplane parameters $\mathbf{w}$ and $b$ that make the samples with a class label $y_i$ of $-1$ have $\mathbf{w}\mathbf{x}_i + b < 0$ and the samples with a class label $y_i$ of $+1$ have $\mathbf{w}\mathbf{x}_i + b > 0$. Mathematically,

finding the hyperplane parameters $\mathbf{w}$ and $b$ that achieve the differentiation can be expressed as an optimization problem. However, a measure (objective function) to differentiate between good hyperplanes and bad hyperplanes (Figure 3.4) needs to be defined.

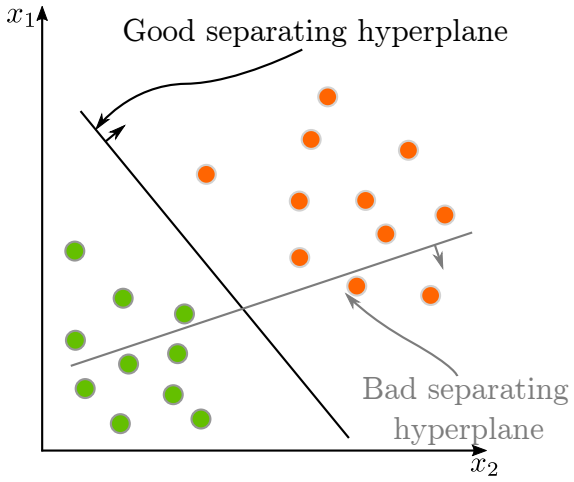


Figure 3.4: Illustration of two hyperplanes, one that separates the data without errors (i.e. "good" hyperplane), and one that doesn't (i.e. "bad" hyperplane ).

Several measures can be used to define this differentiation, such as the misclassification. If misclassification is used then the objective function $f(\mathbf{w}, b)$ can be stated as:

$$f(\mathbf{w}, b) = \sum_{i}^{l} I\{y_i(\mathbf{w}\mathbf{x}_i + b) < 0\}, \tag{3.5}$$

where $I$ is an indicator function that gives zero if the argument is false and one if it is true and $l$ denotes the number of samples. However, this measure is not the best for optimization. Firstly, because it is non-convex [1], which makes the search for the best hyperplane difficult. Secondly, it will make the algorithm produce different results in different executions. To understand this last behaviour consider Figure 3.5. This figure illustrates the misclassification by considering every possible hyperplane. In other words, each point illustrates the misclassification by considering a given hyperplane. As it can be observed from Figure 3.5 there are several hyperplanes (dark blue region delimited by a white line) that can be used to separate the two classes. This implies that different executions of the algorithm can give a different set of parameters of $\mathbf{w}$ and $b$ as the best hyperplane.

Convex objective functions over convex sets are in some "sense" easier to optimize than the general case [10–12]. This property makes them desirable to use as objective

[1] Convexity implies that the union of any two function points by means of a line segment does not intersect the function.

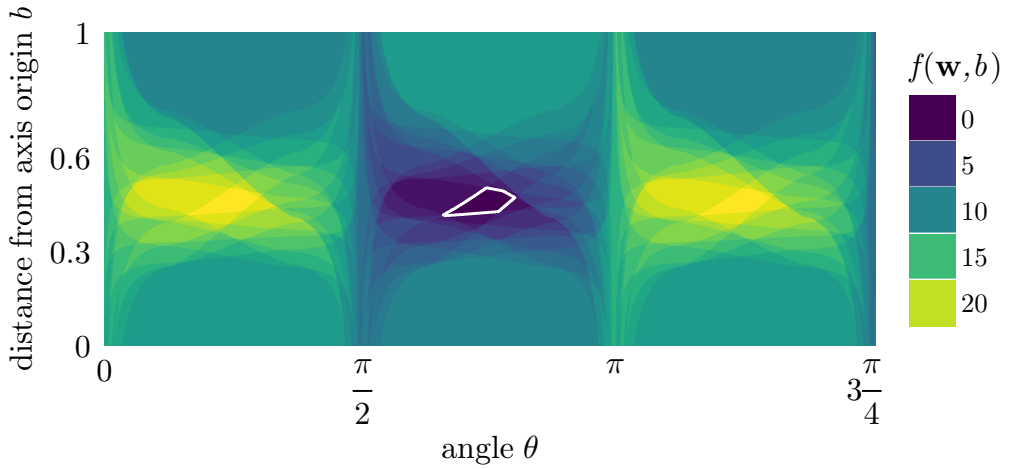


Figure 3.5: Misclassification obtained for the dataset illustrated on Figure 3.4. Each point represents the misclassification for the hyperplane which projected at $y = 0$ exhibits an angle $\theta$ and has a offset from the axis origin of b.

functions. Furthermore, stating the above problem with a convex objective function is straightforward and can be done in a number of ways. However, SVMs consider the objective function whose best achievable hyperplane exhibits maximum margin to the training samples (Figure 3.6) [9]. It turns out that if such hyperplane exists, then an upper bound on the test error can be stated in terms of the training error [13].

Finding the hyperplane that exhibits maximum margin to the training patterns amounts to maximize the distance between the hyperplane and the closest training patterns (support vectors). Denoting the hyperplane distance to the closest negative and positive training patterns by $d_-$ and $d_+$, then this amounts to maximize $d = d_+ + d_-$. It follows that the margin can be described in terms of the hyperplane by considering that the closest positive training pattern exhibits a distance from the origin of $|1 - b| / \|\mathbf{w}\|$ and the closest negative training pattern exhibits a distance from the origin of $|-1 - b| / \|\mathbf{w}\|$, hence $d = 2/\|\mathbf{w}\|$. As a result, the hyperplane that gives maximum margin can be expressed as:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to:} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \geq 0, \forall i \end{aligned} \tag{3.6}$$

For the case illustrated in Figure 3.3, it is to be expected that the solution to the optimization problem has the form illustrated in Figure 3.6. In this figure the support vectors are circled and represent the points for which the equality in Equation (3.6) holds. Note, that the removal of this points would imply a different solution to the optimization problem.

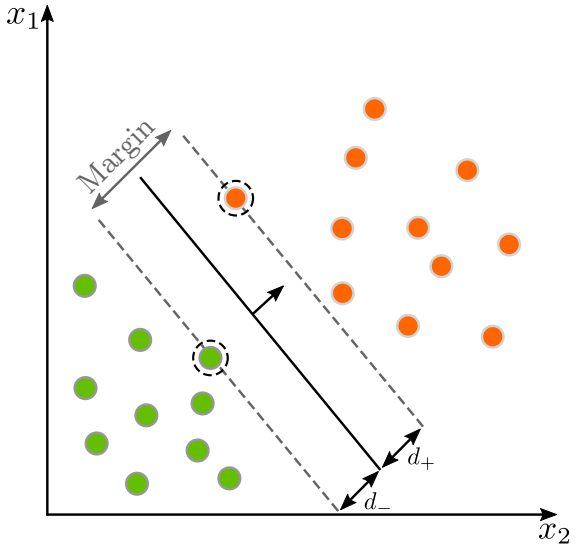


Figure 3.6: Maximum margin linear hyperplane for the case illustrated on Figure 3.3. The support vectors are represented by highlighted by circles.

## The non separable case (outliers)

In the previous subsection, support vector machines were described when the data was linearly separable. However, the above optimization problem (Equation (3.6)) will find no feasible solution when applied to non-separable data. For instance, when there are outliers. The reason is that there will always exist violated constraints. Consequently, one might ask how should the above ideas be extended to handle non-separable data? Since, the constraints are only violated for the non-separable data, one idea would be to relax those constraints. In doing so, one would have to introduce a further cost in the objective function representing the relaxed constraints (Figure 3.7). Mathematically this can be accomplished by introducing positive slack variables $\xi_i, i = 1, \cdots, l$ in the constraints. That is, one for every constrain:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i \geq 0 \tag{3.7}$$

It can be observed, from Equation (3.7) that for an error to occur the corresponding slack variable $\xi_i$ must exceed unity. As a result, $\sum_i \xi_i$ represents an upper bound on the number of training errors. It follows that $\sum_i \xi_i$ expresses a straightforward representation of the cost of violating the constraints [14]. Consequently, the optimization problem can be expressed as:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i^l \xi_i \\ \text{subject to:} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i \geq 0 \\ & \xi_i \geq 0, \forall i \end{aligned} \tag{3.8}$$

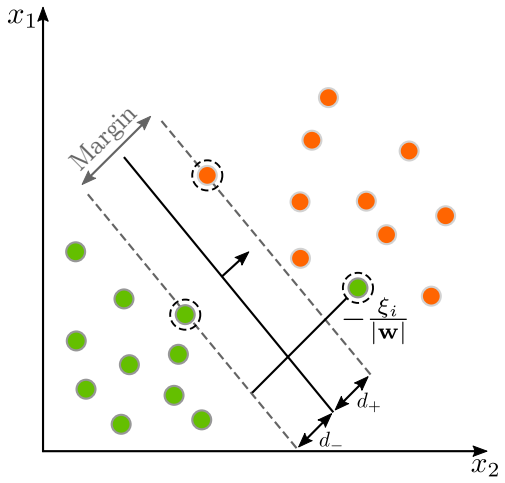


Figure 3.7: Maximum margin separating hyperplane with a penalized outlier.

where $C$ is a weighting term chosen by the user to penalize the errors.

## Non-linear support vector machines

Up until now support vector machines (SVMs) were described considering that the data was linearly separable. However, what if the data is non-linearly separable (i.e. it cannot be separable by a linear function). Can SVMs be adapted to handle this situation as well? Boser *et al.* [15] showed that by using a rather old trick SVMs could be adapted to handle non-linear separable data. Boser idea consists in mapping the data into some higher dimensional feature space $F$ using a function $\phi(\cdot)$ where the data is linearly separable (Figure 3.8).

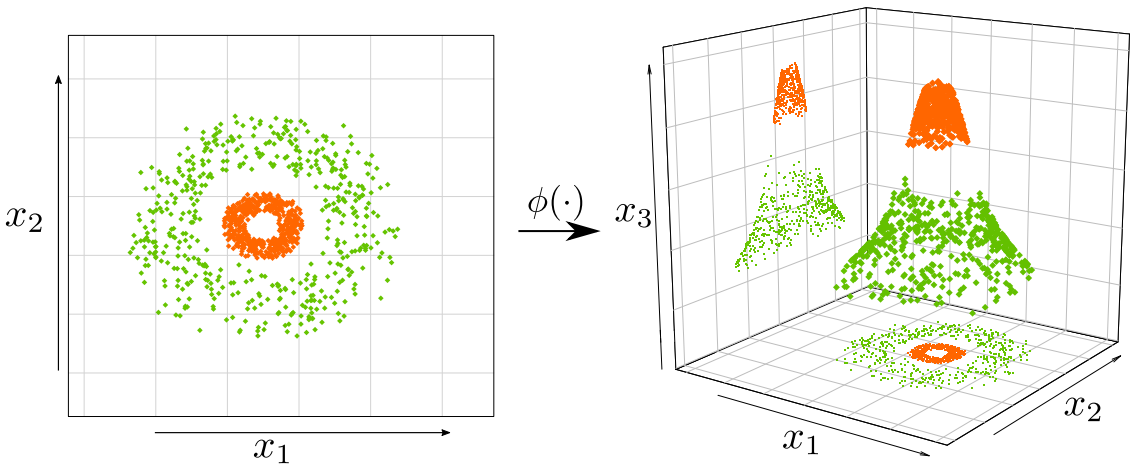


Figure 3.8: a) Illustration of a non linearly separable binary classification problem where the dots represent the samples and colours the different classes. b) The same illustration as in a) mapped onto a high dimensional feature space $F$ using a mapping function $\boldsymbol{\phi}(\cdot)$.

The trick resides in the fact that even-though $\boldsymbol{\phi}(\cdot)$ cannot be easily computed, it can be expressed in terms of inner products between the images of all pairs of data. In other words, there is a function $K(\mathbf{x}_i, \mathbf{x}_j) =< \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) >$. However, the optimization problem (3.8) needs to be expressed in terms of dot products between the images pairs, which is accomplished through the dual Lagrangian formulation [15]. The primal Lagrangian formulation of the problem using the data mapping function $\boldsymbol{\phi}(\cdot)$ is then [15]:

$$\begin{aligned}
\min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i^l \xi_i \\
& - \sum_i \alpha_i\{y_i(\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}_i)+b) - 1 + \xi_i\} - \sum_i \mu_i\xi_i && (3.9)\\
\text{subject to:} \quad & \mathbf{w} = \sum_i \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i) && (3.10)\\
& \sum_i \alpha_i y_i = 0 && (3.11)\\
& C - \alpha_i - \mu_i = 0 && (3.12)\\
& y_i(\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}_i)+b) - 1 + \xi_i \geq 0 && (3.13)\\
& \xi_i \geq 0 && (3.14)\\
& \alpha_i \geq 0 && (3.15)\\
& \mu_i \geq 0 && (3.16)\\
& \alpha_i\{y_i(\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}_i)+b) - 1 + \xi_i\} \geq 0 && (3.17)\\
& \mu_i\xi_i = 0 && (3.18)
\end{aligned}$$

where $\alpha_i$, $\mu_i$ are both Lagrange multipliers introduced to handle both constraints of optimization problem (3.8). The dual formulation of (3.9) is:

$$\begin{aligned}
\max \quad & \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \boldsymbol{\phi}(\mathbf{x}_i)\cdot\boldsymbol{\phi}(\mathbf{x}_j) \\
\text{subject to:} \quad & 0 \leq \alpha_i \leq C \\
& \sum_i \alpha_i y_i = 0
\end{aligned} \qquad (3.19)$$

where $\alpha_i$, $\alpha_j$ are the Lagrange multipliers. Note that the optimization problem (3.19) does not depend on the slack variables $\xi_i$ or the parameters $\mathbf{w}$ and $b$. Furthermore, it can be seen that the user parameter $C$ represents an upper bound on the Lagrange multipliers $\alpha_i$. The optimization problem (3.19) can be solved by replacing the dot product $\boldsymbol{\phi}(\mathbf{x}_i)\cdot\boldsymbol{\phi}(\mathbf{x}_j)$ by a simple kernel evaluation $K(\mathbf{x}_i, \mathbf{x}_j)$ such as the Gaussian kernel [15]:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2} \tag{3.20}$$

However, this raises the question on how to find the parameters $\mathbf{w}$ and $b$ since both reside in $F$. It follows that $b$ can be retrieved by making used of the KKT (Karush-Kuhn-Tucker) conditions [16], which state that at the optimum the product between dual variables and constraints vanishes. Hence, $\alpha_i\{y_i(\mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x}_i) + b) - 1 + \xi_i\} = 0$ and $\mu_i \xi_i = 0$. Considering, Equations (3.12) and (3.18) it can be seen that $\xi_i = 0$ if $\alpha_i < C$. Consequently, $b$ can be found by solving (3.17) for any training sample that satisfies $0 < \alpha_i < C$. However, Equation (3.17) depends on $\mathbf{w}$, and $\mathbf{w}$ cannot be cannot be readily computed because $\boldsymbol{\phi}(\cdot)$ is not known (see Equation (3.10)). Nonetheless, one does not need to. The reason is that, to compute $b$, Expression (3.17) can be written as:

$$\begin{aligned} b &= \frac{1}{y_i} - \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x}_i) \\ &= \frac{1}{y_i} - \sum_{j=1}^{N_s} \alpha_j y_j \boldsymbol{\phi}(\mathbf{s}_j) \cdot \boldsymbol{\phi}(\mathbf{x}_i) \\ &= \frac{1}{y_i} - \sum_{j=1}^{N_s} \alpha_j y_j K(\mathbf{s}_j, \mathbf{x}_i) \end{aligned} \tag{3.21}$$

for any training sample $\mathbf{x}_i$ (whose corresponding constrained $\alpha_i$ verifies $0 < \alpha_i < C$) and to evaluate a test sample $\mathbf{x}$ the decision function can be expressed as the sign of:

$$f(\mathbf{x} = \sum_{j=1}^{N_s} \alpha_j y_j \boldsymbol{\phi}(\mathbf{s}_j) \cdot \boldsymbol{\phi}(\mathbf{x}) + b \tag{3.22}$$

$$= \sum_{j=1}^{N_s} \alpha_j y_j K(\mathbf{s}_j, \mathbf{x}) + b \tag{3.23}$$

where $\mathbf{s}_j$ are the support vectors and $N_s$ the number of support vectors. This procedure, however, can be computationally expensive for problems with a large number of support vectors $\mathbf{s}_j$.

In the previous sections I have discussed the main ideas of binary classification using SVMS. In the next section I briefly discuss how SVMs can be adapted to handle multi-classification data and not only binary classification.

## Multi-classification

Multi-classification is still an open topic in SVM research. Several adaptations to extend support vector machines into multi-classification have been proposed [17].

However, none seems to have been favoured over the more straightforward approaches, one versus all [18] and all versus all [19]. One versus all considers the individual training of binary classifiers, one for each class on the training set, where the positive examples represent the training set class to be discerned and the negative examples represent the remaining training set classes. Classification using this method is just a matter of verifying which of the classifiers gives the trained positive class as a response. All versus all, on the other hand, considers the training of $N(N-1)$ binary classifiers, where each classifier attempts to discern between two training set classes. Classification of test examples on the later is more involved as it requires the evaluation of $N(N-1)$ classifiers responses and verification of which of the different classifiers trained with positive examples of class $i$ had the highest number of votes. Denoting the response vote of a classifier trained with positive examples of class $i$ and negative examples of class $j$ by $r_{ij}$, then the class $c$ is determined by evaluating the following expression:

$$c(\mathbf{x}) = \underset{i}{\operatorname{argmax}} \sum_{j} r_{ij}(\mathbf{x}). \tag{3.24}$$

In practice only $N(N-1)/2$ classifiers are required to be trained since $r_{ji} = -r_{ij}$. Specifically, in support vector machines "all versus all" was found to be a competitive approach [17] and as a result is often used.

## One-class support vector machines

Support vector machines assume the existence of two categories to build a descriptive model. This assumption was quite successful for obtaining a multi-class classifier as discussed in previous sections. In this section I show how SVMs were extended to learn a descriptive model from a single category – one class learning. At the time of this writing I am aware of two SVM extensions that build a descriptive model from one category, Schölkopf *et al.* [20] and Tax and Duin [21]. In both approaches it is assumed that anomalies are not concentrated. However, they differ on the method to obtain the closed class boundary of concentrated normal data.

Schölkopf *et al.* consider the origin as the only member of the anomaly class, $-1$, and aim to separate the normal class data from the origin with maximum margin. Tax and Duin, on the other hand, assume a spherical boundary encompassing the normal class data $+1$, and aim to minimize the hypersphere volume so that the probability of including outliers is minimized. These assumptions led to the optimization problems described below. Note that in the description of the optimization problems a set of $d$ training examples $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, ..., $(\mathbf{x}_d, y_d) \in \mathbb{R}^n$ is assumed.

### Schölkopf approach to one-class learning

In Schölkopf *et al.* approach the hyperplane that separates the training examples with maximum margin from the origin can be obtained by solving the following optimization problem (dual Lagrangian formulation) [20]:

$$\begin{aligned} \max \quad & -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j K(\mathbf{x}_i,\mathbf{x}_j) \\ \text{subject to:} \quad & 0 \leq \alpha_i \leq \frac{1}{vd} \\ & \sum_i \alpha_i = 1 \end{aligned} \tag{3.25}$$

where $\alpha_i$, $\alpha_j$ are the Lagrange multipliers, $K(\mathbf{x}_i,\mathbf{x}_j) =< \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) >$ the kernel function, $d$ the number of training examples and $v$ a regulation parameter that controls the trade-off between the number of allowed outliers and how small the regularization term $\|\mathbf{w}\|$ can be. Once solved, the above optimization problem will yield the following decision function:

$$f(\mathbf{x}) = \text{sign}(\sum_i \alpha_i K(\mathbf{x}_i,\mathbf{x}) - b). \tag{3.26}$$

The decision function (3.26) is expressed in terms of the Lagrange multipliers $\alpha_i$, the training patters $x_i$ (support vectors) and $b$. The parameter $b$ can be obtained, as in previous sections, by exploiting the fact that, at the optimum, any $\alpha_i$ that satisfies $0 < \alpha_i < \frac{1}{vd}$ has a corresponding training pattern $\mathbf{x}_i$ that is a support vector, hence:

$$b = (\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}_i)) = \sum_j \alpha_j K(\mathbf{x}_j,\mathbf{x}_i) \tag{3.27}$$

Finally, the classification of new patterns $\mathbf{x}$ is just a matter of verifying the sign of the decision function.

### Tax and Duin approach to one-class learning

Tax and Duin assume that normal data is enclosed in an hypersphere of radius $R$ and centre $\mathbf{a}$. The aim is to minimize the hypersphere volume $R^2$ such that all training patterns $\mathbf{x}_i$ are enclosed in it. In mathematical terms this implies the following optimization problem [21]:

$$\begin{aligned} \min \quad & R^2 + C\sum_i \xi_i \\ \text{subject to:} \quad & \|\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{a}\|^2 \leq R^2 + \xi_i \\ & \xi_i \geq 0 \quad \forall i \end{aligned} \tag{3.28}$$

where $\xi_i$ are slack variables introduced to penalize outliers, i.e. samples where the distance from the hypersphere origin $\mathbf{a}$ exceeds $R^2$, $\boldsymbol{\phi}(\cdot)$ a function that maps data onto some high dimensional feature space $F$ and $C$ a parameter that controls the trade-off between the volume and the number of outliers. To solve the optimization problem (3.28) the Lagrangian formulation is used and can be stated in terms of its dual as:

$$\begin{aligned} \max \quad & \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to:} \quad & 0 \leq \alpha_i \leq C \end{aligned} \tag{3.29}$$

where $\alpha_i$ are the Lagrange multipliers and $K(\mathbf{x}_i, \mathbf{x}_j) =< \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) >$ the kernel function. Solving optimization problem (3.29) will give $\mathbf{a}$ and $R$. The centre of the sphere $\mathbf{a}$ is given by the primal Lagrangian formulation and is a linear combination of the support vectors (training samples for which the Lagrange multiplier is non-zero and smaller than $C$) $\mathbf{a} = \sum_i \alpha_i \boldsymbol{\phi}(\mathbf{x}_i)$. The radius of the sphere $R$ can be obtained by evaluating the distance of any of the support vectors $\mathbf{x}_k$ to the centre of the sphere, hence:

$$\begin{aligned} R^2 &= \|\boldsymbol{\phi}(\mathbf{x}_k) - \mathbf{a}\|^2 \\ &= \|\boldsymbol{\phi}(\mathbf{x}_k)\|^2 - 2\mathbf{a} \cdot \boldsymbol{\phi}(\mathbf{x}_k) + \|\mathbf{a}\|^2 \\ &= K(\mathbf{x}_k, \mathbf{x}_k) - 2\sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_k) + \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \end{aligned} \tag{3.30}$$

To test a new pattern $\mathbf{x}$ the distance to the centre of the sphere $\mathbf{a}$ has to be calculated. The pattern is considered normal if the distance is smaller than or equal to the radius $R$ and abnormal if greater:

$$\|\mathbf{x} - \mathbf{a}\|^2 = K(\mathbf{x}, \mathbf{x}) - 2\sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \leq R^2 \tag{3.31}$$

## 3.3 Final Remarks

Random Forests (RF) are inherently a multi-classification algorithm that requires little or no parameter tuning to perform well for balanced training data. However, when the number of examples of one of the classes greatly exceeds the others (i.e. training data is unbalanced) classification performance is affected. The reason is that the highest frequent class will figure with a higher representativeness in any of the bagged training sets and will affect the node splitting process. In other words, class representativeness will not be taken into account for node splitting resulting in a skew towards the highest represented class. Two actions can be used

to mitigate this effect. The first is to modify the bagging process to generate training datasets with the same class representativeness. The second is to incorporate class representativeness in the node splitting measure. In either case, the class division will improve but it will still be dependent on the available data.

Support Vector Machines (SVM), on the other hand, are a binary classification algorithm. Multi-classification in SVMs is achieved through the combination of several SVM models. Hence, from this point of view only, SVMs are a technique inherently more complex than RF. However, SVMs should not be influenced by unbalanced training data. The reason is that class division in SVMs is only dependent on the support vectors. Hence, if one of the categories has a higher representativeness than the other but only a small percentage are support vectors class boundary division will not be severely affected. Nonetheless, since very few problems are linearly separable, performance in SVMs is mostly influenced by the choice of Kernel and regulation parameter value. And while both can be chosen using a cross validation strategy during the training of multi-class problems, the same is not true for one-class SVMs. Choosing the right Kernel for a specific anomaly detection problem is often a question of the researcher experience with the problem and is still the matter of ongoing research in the SVM field.

Despite the advantages and disadvantages of each of the methods their applicability will ultimately depend on the problem.

As a side note, and for the interested reader, I would like to empathize that I made available the implementations on javascript of random forests and support vector machines for classification tasks on my github page [22].

## 3.4 Bibliography

[1] Usama Fayyad, Gregory Piatetsky-shapiro, and Padhraic Smyth. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17:37–54, 1996.

[2] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, New York, NY, USA, 2009.

[4] Leo Breiman, Jerome Friedman, Charles J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Taylor & Francis Ltd, 1984. ISBN 0412048418.

[5] L. Rokach. *Data Mining with Decision Trees: Theory and Applications*. Series in machine perception and artificial intelligence. Rokach, Lior, 2008. ISBN 9789812771728.

[6] Max Bramer. *Principles of Data Mining*. Springer London, 2013. doi: 10.1007/978-1-4471-4884-5.

[7] Yali Amit and Donald Geman. Shape Quantization and Recognition with Randomized Trees. *Neural Computation*, 9(7):1545–1588, October 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.7.1545.

[8] Christopher J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, June 1998. ISSN 1384-5810. doi: 10.1023/A:1009715923555.

[9] V. Vapnik and A. Lerner. Pattern Recognition using Generalized Portrait Method. *Automation and Remote Control*, 24, 1963.

[10] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

[12] Andrzej Ruszczynski. *Nonlinear Optimization*. Princeton University Press, Princeton, NJ, USA, 2006. ISBN 0691119155.

[13] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

[14] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995. ISSN 1573-0565.

[15] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X.

[16] R. Fletcher. *Practical Methods of Optimization; (2Nd Ed.)*. Wiley-Interscience, New York, NY, USA, 1987. ISBN 0-471-91547-5.

[17] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, Mar 2002. ISSN 1045-9227.

[18] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of classifier methods: a case study in handwritten digit recognition. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1994. Vol.*

*2 - Conference B: Computer Vision amp; Image Processing.*, volume 2, pages 77–82, Oct 1994.

[19] S. Knerr, L. Personnaz, and G. Dreyfus. *Single-layer learning revisited: a stepwise procedure for building and training a neural network*, pages 41–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 1990. ISBN 978-3-642-76153-9.

[20] B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, pages 582–588, 2000.

[21] David M. J. Tax and Robert P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20:1191–1199, 1999.

[22] B. F. Faria. Random forests and support vector machines javascript demo implementations, 2016. URL `https://github.com/BrunoFFaria/Data-mining`.

CHAPTER 4

# Cellular Frustration Algorithm

The Cellular Frustration Model (CFM) is an alternative view of how the immune system performs pathogen detection. Instead of considering that detection is the result of the individual and independent behaviour of each cell as the Negative Selection Algorithm (NSA) does, the CFM considers that it emerges from the correlated action of all constituents. The different considerations requires both models to define mechanisms accommodating an high reactivity against foreign elements with a low reactivity (tolerance) against elements from the body differently. NSA seeks to restrict the individual T cells action domain by not containing any self elements (elements from the body). However, as already discussed this has the disadvantage that perfect discrimination is unreachable. On the other hand, the CFM uses the constituents interaction lifetime to define reactivity and tolerance which is explained in the next sections.

The aim of the next sections is to describe the CFM as it was before this thesis. I describe the main idea behind the CFM, how it achieves tolerance and reaction, and how this idea has been materialized in an algorithm to perform self-nonself discrimination.

## 4.1 Cellular frustration concept

The biggest conceptual change introduced by the CFM is that during an interaction the involved cells use some form of decision process to change state which takes time [1–3]. Tolerance and reaction within this view depends solely on the time cells spent interacting. Cells that continuously interact over a predefined period of time trigger a reaction, while cells that continuously change state remain in a toleration state. To better convey how this process develops to the reader, a simple metaphor involving men and women is used. In this metaphor men and women purport the

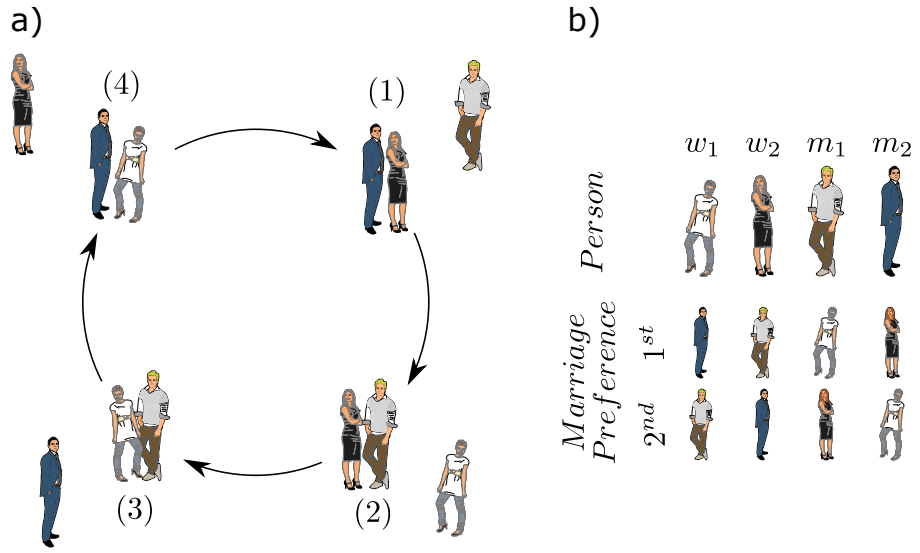


Figure 4.1: a) Illustration of the agents dynamics. b) The agents' marriage preference ordering. Note that a population of agents in this configuration will never reach a stable solution, i.e. there will never be a marriage that cannot be terminated.

roles of antigen presenter cells (APCs) and T cells, and are assumed to have an ordering of preference for the opposed genre. Consider then, a set of 2 men and a set of 2 women with ordering of preferences as defined in Figure 4.1, where: man 1 ($m_1$) prefers woman 1 ($w_1$) to woman 2 ($w_2$); man 2 ($m_2$) prefers woman 2 ($w_2$) to woman 1 ($w_1$); woman 1 prefers man 2 to man 1 and woman 2 prefers man 1 to man 2.

Starting with the first marriage illustrated on Figure 4.1 by the marriage between $m_2$ and $w_2$ and designated by (1). If $m_1$ proposes to $w_2$, then $w_2$ will face a decision, either she continues married with $m_2$ or divorces $m_2$ and marries $m_1$. It follows that $w_2$ prefers to be married with $m_1$ than with $m_2$. Consequently, $w_2$ will divorce $m_2$ and marry $m_1$ forming the marriage (2). However, in the newly formed marriage $w_1$ can propose to $m_1$, provoking a divorce and forming a new marriage. The cycle can go on as depicted in Figure 4.1, and shows the effect of frustration: that any formed marriage on this conditions will be short lived.

The next situation to be analysed is the situation where a strange woman $s$, $w_s$, appears on the population. Consider that this woman has a goal and in order to reach it she has to pretend to be woman 1. Consider also that the strange woman knows how to pose as woman, i.e. from the men point of view she is woman one but does not know woman 1 ordering of preference for men. Consequently, it has to choose an ordering of preference. In this simple example with only two agents the strange woman only has two different possible orderings, either she chooses woman 1 or woman 2 ordering of preference for men, however in populations with greater number of agents the number of possible orderings can be quite large. Assume

that the strange woman chooses woman 1 ordering of preference for men. In this scenario the dynamics will not be perturbed as nothing has been changed from the scenario depicted on Figure 4.1. However, if the strange woman chooses woman 2 ordering of preference for men, then the cycle on Figure 4.1 is broken as the step from marriage (3) to marriage (4) ceases to exist. Consequently, marriage (3) will be long lived when compared to any marriage when there is no strange woman. It is this difference on the liveness time of marriages that forms the basic argument for pathogen detection according to the CFM.

The watchful reader may argue that for this hypothesis to work there must always exist unengaged men and women. In fact, if the above example started with a configuration where all men and women were already engaged, then no couple changes would take place. This is the situation of stable matching [4]. However, one could argue that in a more general analysis, by considering a higher number of agents, this is an unlikely scenario and as a result the main outcomes presented above will not change. These ideas have been used by the main supervisor of this thesis to propose an alternative view (CFM) on how the human adaptive immune system is triggered.

## 4.2 Cellular frustration model

In the CFM instead of men and women there are two cell types, APCs and T cells, that continuously engage in a decision dynamics process. APCs scavenge the body looking for antigen. Once found they migrate to the lymph nodes where they present the antigen (information) to T cells. Depending on the information presented, T cells establish contacts with different strength and times. In particular, when nonself antigen is detected on the surface of APCs, T cells develop longer and stronger contacts [5, 6]. Thus, I start by defining the main building block of the CFM, the agent:

**Definition 4.1.** *(**Agent**) An agent $a_i \in A$, with index $i \in \{1, 2, \ldots, N\}$, $N = |A|$, is defined as a tuple $a_i = \{l_i, L_i, K_i, s_i\}$, where:*

- $l_i \in X \subset \mathbb{N}$ *is the ligand;*
- $K_i$ *is the set of all agents that agent $a_i$ can interact with;*
- $L_i$ *is an ordered sequence of ligands and is denoted as the list or receptor;*
- $s_i$ *is the pairing state of agent $a_i$, and holds the index of the agent to which the agent $a_i$ is paired or zero if it is unpaired;*

The set of agents $A$ is divided in two types, presenters and detectors, that play the role of APCs and T cells, respectively. Hence, I introduce the following definition to describe the agents in $A$:

**Definition 4.2.** *The set of agents $A$ is formed by two subsets (or types) of agents, presenters, $P$, and detectors, $D$, such that $A = P \cup D$ and $P \cap D = \emptyset$.*

From Definition 4.2 the number of agents in the set $A$ ($N$) is given by $N = N_P + N_D$, where $N_P$ denotes the number of presenters in $P$, i.e. $N_P \equiv |P|$, and $N_D$ the number of detectors in $D$ ($N_D \equiv |D|$). For simplicity in this work it is assumed that $N_D = N_P$.

The next definition concerns the ligands displayed by each agent type. In the immune system, APCs present a huge variety of different ligands which derive from the diversity of the captured antigen. In comparison, T cells exhibited ligands are considered to be much less diverse and can result for instance from the cells' surface molecules. This considerations led to the following definition for the agent's exhibited ligand:

**Definition 4.3.** *In the cellular frustration model, the set of ligands displayed by detectors, $X_D$, is given by $X_D = \{l_i : l_i \in \{1, 2\} \wedge a_i \in D\}$. The set of ligands displayed by presenters, $X_P$, is given by $X_P = \{l_i : l_i \in X \subset \mathbb{N} \wedge a_i \in P\}$.*

In the CFM it is assumed, for simplicity, that agents of the same type do not interact. Consequently, the following definition is used to define each agent's set of possible interactions, $K_i$:

**Definition 4.4.** *In the cellular frustration model interactions involve only agents of different types, so that, $K_i \subseteq P, \forall a_i \in D$ or $K_i \subseteq D, \forall\ a_i \in P$.*

A direct consequence of Definitions 4.3 and 4.4 is that presenters exhibit only two different receptor lists given by the ordered tuples $(1, 2)$ and $(2, 1)$. By contrast, detectors lists are much more diverse since they encode the orderings of the ligands exhibited by presenters. This implies that a further distinction can be made at the level of each agent type:

**Definition 4.5.** *In the cellular frustration model, detectors and presenters can be divided in two subtypes. Detectors with $l_i = 1$ belong to subtype 1, while detectors with $l_i = 2$ belong to subtype 2. Conversely, presenters with $L_i = (1, 2)$ belong to subtype 1, while presenters with $L_i = (2, 1)$ belong to subtype 2.*

A simple representation of the model can be appreciated in Figure 4.2. In this work the different subtypes are considered to have an equal number of agents.

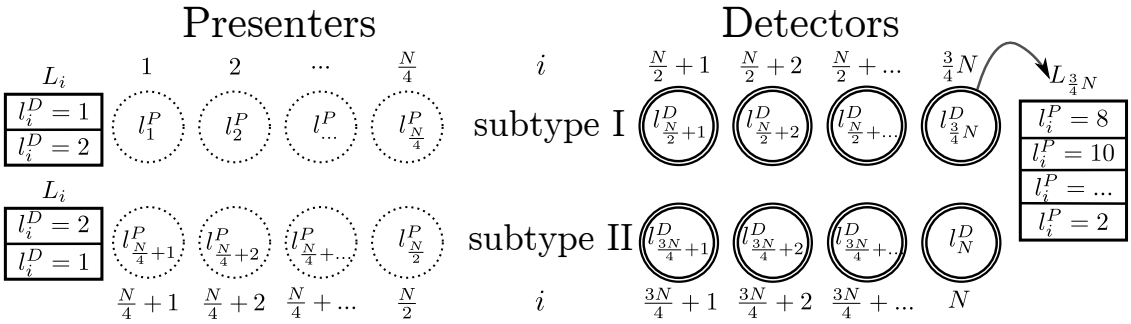


Figure 4.2: A schematic representation of the cellular frustration model with equal number of presenters and detectors and an equal number of agents in each subtype. Presenters from subtype 1 prefer to pair with detectors exhibiting the ligand 1, while presenters from subtype 2 prefer to pair with detectors exhibiting ligand 2. Detectors lists are ordering of preferences for all ligands that can be displayed by presenters.

The next definition concerns the agents' pairing state. In the CFM it is assumed that agents of the same type do not interact. Hence, agents can only be paired with agents of the opposite type. Furthermore, it is also assumed that agents cannot be paired with more than one agent. For instance, any time an agent tries to interact with a paired agent, both interacting agents face a decision: either they establish a new pairing and terminate the former pairings, or they stay in the former pairs. The agents' pairing state is update through a set of rules that are denoted as decision rules. If $r_l(s)$ represents the rank of a string $s$ on list $l$, then the following decision rules can be defined:

**Definition 4.6.** *(**Decision Rules**) The following set of decision update rules R, can be defined when agents $a_i \in D$ and $a_j \in P$ interact:*

*I* *if* $s_i = 0 \wedge s_j = 0$ *then* $s_i \to j, s_j \to i$.

*II* *if* $s_i = k \wedge s_j = 0 \wedge r_{L_i}(l_j) < r_{L_i}(l_k)$ *then* $s_i \to j, s_j \to i, s_k \to 0$.

*III* *if* $s_j = k \wedge s_i = 0 \wedge r_{L_j}(l_i) < r_{L_j}(l_k)$ *then* $s_j \to i, s_i \to j, s_k \to 0$.

*IV* *if* $s_i = k \wedge s_j = p \wedge r_{L_i}(l_j) < r_{L_i}(l_k) \wedge r_{L_j}(l_i) < r_{L_j}(l_p)$ *then* $s_j \to i, s_i \to j, s_k \to 0, s_p \to 0$.

Figure 4.3 illustrates a particular example of how Definition 4.6 is applied to create a new pairing and a new unpaired agent.

A dynamical system is established when agents are put in interaction and decision rules are applied to change agents' states. Different algorithms can be defined to put agents in interaction. However, the CFM assumes that agents interact following random encounters, an assumption that agrees with immunological

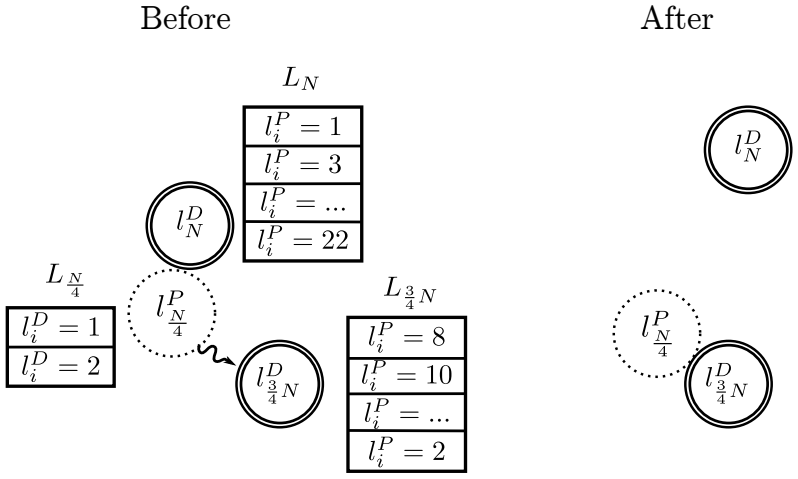


Figure 4.3: Example of application of decision rule (II) in Definition 4.6. In this example, a detector $a_i$, with list $L_i$, is paired with a presenter $a_k$, displaying ligand $l_k^P$. Detector $a_i$ interacts with presenter $a_j$ displaying ligand $l_j^P$. Since ligand $l_j^P$ is ranked first in $L_i$, a new pairing is formed by agents $a_i$ and $a_j$ and agent $a_k$ becomes unpaired (i.e., $s_k$ is set to 0).

observations [6, 7]. Consequently, the following definition of stochastic iteration is an acceptable approximation for this assumption:

**Definition 4.7.** *(**Stochastic Iteration**) Consider a sequence with all agents. During a stochastic iteration all agents in this sequence interact sequentially with a randomly picked agent of the other type and agents' pairing states are updated following decision rules in Definition 4.6. In the end, the stochastic iteration number is incremented.*

Definition 4.7 has two implications. First, several agents could interact with a same agent in the same interaction. Second, it defines an ordering for the interactions. However, both implications are of no consequence in the duration of a pairing, which only takes into consideration how agent's states change along iterations. In this respect, it is important to first define the configuration of a population at the end of an iteration:

**Definition 4.8.** *(**Configuration**) A configuration of the population $\mathcal{C}$, is given by the set of unpaired agents plus the set of pairings:* $\mathcal{C} = \{a_k : s_k = 0, k = 1, \ldots, N\} \cup \{(a_k, a_m) : s_k = m \wedge s_m = k, k, m = 1, \ldots, N\}$

The duration of a pairing $M_{ij}$ involving agents $a_i$ and $a_j$ can then be defined as:

**Definition 4.9.** *(**Pairing Duration**) The duration of a pairing $M_{ij}$ formed by agents $a_i$ and $a_j$ is equal to the number of stochastic iterations between two consecutive updates of these agents states $s_i$ and $s_j$.*

The next topic to be addressed is how the CFM organizes the dynamics in face of the information displayed by presenters. The CFM achieves dynamical organization through detector selection. This is because the number of possible detectors, due to the extremely varied detector lists, is extremely large. Hence, to achieve a desired dynamical organization a repertoire of detectors has to be selected. In the CFM detector selection mimics the immune system T cell thymic selection, in which inadequate T cells are eliminated, by replacing detectors with inadequate lists by other detector agents. Thus negative selection is defined as follows:

**Definition 4.10.** *(**Negative Selection**) Negative selection replaces the list of the detector agents $a_j$ having pairing durations $M_{ij}$ larger than a negative threshold $\tau_{NS}$ by a new randomly ordered list.*

The negative selection threshold is not necessarily constant during the thymic education process. In the CFM it is defined as:

**Definition 4.11.** *(**Negative selection adaptive threshold**) In repertoire education with adaptive threshold, $\tau_{NS}$ is updated to the largest pairing lifetime in the last $W \in \mathbb{N}$ iterations if no pairing involving detector agent $a_j$ lasted longer than $\tau_{NS}$.*

The education process applies Definitions 4.10 and 4.11 to a population of detectors during $E$ iterations with $E >> W$ and $E \in \mathbb{N}$. Every time $\tau_{NS}$ is updated the population of detectors is saved. A repertoire of selected detectors is then obtained by repeating this procedure $N_D^{rep}$ times, one for each population of detectors. After which, the population enters in the final stage of repertoire education. This stage, denoted as calibration, aims to establish the default pairing durations distribution for each presenter. To accomplish this, the agents engage in the same decision dynamics as before, except now anergy is introduced:

**Definition 4.12.** *(**Anergy**) Anergy terminates the pairings involving agents $a_i$ and $a_j$ paired for a time longer than $\tau_A$ and replaces the detector agent of the terminated pairing by detector agents with the same set of possible agent interactions $K_i$ in the repertoire.*

An important hypothesis used by the CFM is that pairing lifetimes are robust anomaly detections indicators [3]. Therefore, it is not so much the pairing duration, as discussed in the previous section, but rather the lifetime - i.e. the slope of the pairing durations distribution - that should be closely monitored. Consequently, an important concept for defining the pairing duration frequency of an agent $a_i$ is the distribution of pairing durations involving agent $a_i$:

**Definition 4.13.** *(**Pairing duration distribution**) The pairing duration distribution $F_i(\tau)$ is equal to the number of pairings involving agent $a_i$ that in the last $W$ iterations had a duration $M_{ij}$ between two consecutive updates of the agent $a_i$ state of $\tau_{act}$ iterations.*

Each agent $a_i$ pairing duration frequency $f_i(\tau_{act})$ is then defined from the pairing distribution as:

$$f_i(\tau_{act}) = \frac{\sum_{j=\tau_{act}}^{\tau_A} F_i(j)}{\sum_{j=0}^{\tau_A} F_i(j)} \tag{4.1}$$

The next and last step of the model is monitoring. In monitoring, the model supervises a set of presenters and triggers a response for presenters that exceed the default pairing duration frequency. More formally, monitoring in the CFM is defined as:

**Definition 4.14.** *(**Monitoring**) In monitoring all agents are put in interaction with anergy during $W \in \mathbb{N}$ iterations and the number of pairings lasting longer than an activation time $\tau_{act}$, involving presenter with index $i$, $F_i^{mon}(\tau_{act})$ are recorded. Presenters whose $f_i^{mon}(\tau_{act})$ exceed the frequency of conjugations lasting longer than $\tau_{act}$ found at the end of repertoire education, $f_i^{edu}(\tau_{act})$, trigger a response.*

In practice however, better discrimination is achieved if the pairing duration frequency found at the end of repertoire education is not strictly enforced [3]. Consequently, a response is triggered for every agent whose $f_i^{mon}(\tau_{act})$ exceeds $f_i^{edu}(\tau_{act}) \times v$, with $v \in [0, 1]$. In the next section I describe how all these definitions fit together to form a self-nonself discrimination algorithm.

## 4.3 Algorithm

A standard implementation of the model described in the previous section is illustrated on Algorithms 4.1, 4.2 and 4.3. The pseudo-code describes the model, as it was before this thesis, for dealing with self-nonself. To complement the pseudo-code I dedicate the next paragraphs to briefly describe what each algorithm does and which model definitions are used.

The CFA is divided in two stages. A first stage that builds a detector repertoire, which is responsible for generating detectors that achieve small pairing lifetimes with presenters exhibiting self information, and a second stage that monitors presenters looking for nonself information. In either stage the algorithm starts by defining the agents informations that is not subject to change. In the case of presenters all the agent information (Definitions 4.1, 4.3, 4.4 and 4.5), whereas in the case

**Algorithm 4.1** The Cellular Frustration Algorithm (CFA).

1: **function** CFA(*educate*, $E$, $W$, $N_D^{rep}$, $R_D$, $l$, $\tau_A$, $\tau_{act}$)
2: $N_D = N_P =$length($l$)
3: $P \leftarrow \emptyset$, $D \leftarrow \emptyset$
4: **for** $i$ in 1 to $N_P$ **do**
5: $a_i.l \leftarrow l_i$, $a_i.K \leftarrow [1, N_D]$, $a_i.s \leftarrow 0$
6: **if** $i <$ length($l/2$) **then**
7: $a_i.L \leftarrow \{1, 2\}$
8: **else**
9: $a_i.L \leftarrow \{2, 1\}$
10: **end if**
11: $P \leftarrow P \cup a_i$
12: **end for**
13: **for** $j$ in 1 to $N_D$ **do**
14: $a_j.K \leftarrow [1, N_P]$, $a_j.s \leftarrow 0$
15: **if** $j <$ length($l/2$) **then**
16: $a_j.l \leftarrow 1$
17: **else**
18: $a_j.l \leftarrow 2$
19: **end if**
20: $D \leftarrow D \cup a_j$
21: **end for**
22: **if** *educate* is *true* **then**
23: $R_D \leftarrow$RepEducation($E$, $W$, $N_D^{rep}$, $P$, $D$, $N_P$, $N_D$)
24: $\{R_D, f^{edu}\} \leftarrow$ CFA(*false*, $E$, $W$, $N_D^{rep}$, $R_D$, $l$, $\tau_A$, $\tau_{act}$)
25: **return** $\{R_D, f^{edu}\}$
26: **else**
27: **for** $j$ in $D$ **do**
28: $k \leftarrow$ random integer between 1 and $N_D^{rep}$
29: $a_j.L \leftarrow R_{D_{(k+j)\times N_D}}.L$
30: **end for**
31: $\{R_D, f^{mon}\} \leftarrow$DynWAnergy($W$, $N_D^{rep}$, $R_D$, $P$, $D$, $\tau_A$, $\tau_{act}$)
32: **return** $\{R_D, f^{mon}\}$
33: **end if**
34: **end function**

of detectors all information except the list. Note that all agents start unpaired and that presenters ligands are assigned from the $l$ vector. Algorithm 4.1, then either builds a detector repertoire or monitors presenters depending on the state of the *educate* variable. If the *educate* variable is set to *true* the algorithm calls the repertoire education (Algorithm 4.2) to build the detector repertoire and then calls itself with *educate* set to *false* to obtain the education pairing frequency $f^{edu}$. If the *educate* variable is *false* the algorithm randomly selects detector lists from the detector repertoire $R_D$ to fill the detector agents lists and then starts the monitoring

stage (Algorithm 4.3).

**Algorithm 4.2** Repertoire education stage of CFA

**function** RepEducation($E$, $W$, $N_D^{rep}$, $P$, $D$, $N_P$, $N_D$)
  $R_D \leftarrow \emptyset$
  **for** $r$ in 1 to $N_D^{rep}$ **do**
    $\forall a_j \in D$, $a_j.L \leftarrow$ random ordering of presenters ligand space
    $D^{saved} \leftarrow \emptyset$
    $\tau_{NS} \leftarrow W$, $num\_reps \leftarrow 0$, $M_{ij} \leftarrow 0$, $\forall i \in [1, N_P]$ and $j \in [1, N_D]$
    **for** $e$ in 1 to $E$ **do**
      $\tau_{NS}^W \leftarrow 0$, $num\_reps \leftarrow 0$
      **for** $w$ in 1 to $W$ **do**
        **for all** $a_i$ in $P \cup D$ **do**
          $a_j \leftarrow$ random agent from $a_i.K$
          $a_i.s^{old} \leftarrow a_i.s$, $a_j.s^{old} = a_j.s$
          Apply decision rules (Definition 4.6)
          **if** $a_i.s \neq a_i.s^{old}$ **then**
            $k = a_i.s^{old}$
            $M_{ik} \leftarrow 0$, if $k \neq 0$
          **end if**
          **if** $a_j.s \neq a_j.s^{old}$ **then**
            $p = a_j.s^{old}$
            $M_{pj} \leftarrow 0$, if $p \neq 0$
          **end if**
        **end for**
        **for all** $a_j$ in $D$ **do**
          $\tau_{NS}^W \leftarrow M_{ij}$, if $M_{ij} > \tau_{NS}^W$
          **if** any $M_{ij} \geq \tau_{NS}, \forall i \in [1, N_P]$ **then**
            $a_j.L \leftarrow$ random ordering of presenters ligand space
            Separate $a_i$ from $a_j$ and set $M_{ij}$ to zero
            $num\_reps \leftarrow num\_reps + 1$
          **end if**
        **end for**
        Increment the $M_{ij}$ of all pairings
      **end for**
      **if** $num\_reps$ is 0 **then**
        $\tau_{NS} \leftarrow \tau_{NS}^W$, if $\tau_{NS}^W < \tau_{NS}$
        $D^{saved} \leftarrow D$
      **end if**
    **end for**
    $R_D \leftarrow R_D \cup D^{saved}$
  **end for**
  **return** $R_D$
**end function**

The repertoire education algorithm aims to build a detector repertoire consisting of $N_D^{rep}$ detector populations that exhibit small pairing lifetimes with presenters

displaying self information. To educate a population of detectors the Algorithm 4.2 starts by setting the detector agents lists with random orderings of the presenters ligand space. Then, negative selection with adaptive threshold (Definitions 4.10 and 4.11) is performed during $E$ stochastic iterations (Definition 4.7). If after $E$ iterations the largest pairing duration is inferior to $W$, the detector population is added to the detector repertoire. Otherwise, the algorithm continues to the next population. In this stage it is convenient to choose $E$ such that after $E$ iterations the largest population pairing duration is small when compared to $W$. The simulations performed on this work often use $W = 10^4$ and $E = 10^6$, unless stated otherwise.

**Algorithm 4.3** Monitoring stage of the CFA.

**function** DynWAnergy($W$, $N_D^{rep}$, $R_D$, $P$, $D$, $\tau_A$, $\tau_{act}$)
  $M_{ij} \leftarrow 0$, $\forall i \in [1, N_P]$ and $j \in [1, N_D]$
  $F_i(\tau) \leftarrow 0$, $\forall i \in [1, N_P]$, $\tau \in [0, \tau_A]$, $N_D = N_P$ =length($D$)
  **for** $w$ in 1 to $W$ **do**
    **for all** $a_i$ in $P \cup D$ **do**
      $a_j \leftarrow$ random agent from agent $a_i.K$
      $a_i.s^{old} \leftarrow a_i.s$, $a_j.s^{old} = a_j.s$
      Apply decision rules (Definition 4.6)
      **if** $a_i.s \neq a_i.s^{old}$ **then**
        **if** $(k = a_i.s^{old}) \neq 0$ **then**
          $F_i(M_{ik}) \leftarrow F_i(M_{ik}) + 1$, $M_{ik} \leftarrow 0$
        **end if**
      **end if**
      **if** $a_j.s \neq a_j.s^{old}$ **then**
        **if** $(p = a_j.s^{old}) \neq 0$ **then**
          $F_p(M_{pj}) \leftarrow F_p(M_{pj}) + 1$, $M_{pj} \leftarrow 0$
        **end if**
      **end if**
    **end for**
    **for all** $a_j$ in $D$ **do**
      **if** any $M_{ij} \geq \tau_A, \forall i \in [1, N_P]$ **then**
        $k \leftarrow$ random integer between 1 and $N_D^{rep}$
        $F_i(\tau_A) \leftarrow F_i(\tau_A) + 1$, $a_j.L \leftarrow R_{D_{(k+j)\times N_D}}.L$
        Separate $a_i$ from $a_j$ and set $M_{ij}$ to zero
      **end if**
    **end for**
    Increment the $M_{ij}$ of all pairings
  **end for**
  **for all** $a_i \in P$ **do**
    $f_i(\tau_{act}) = \frac{\sum_{j=\tau_{act}}^{\tau_A} F_i(j)}{\sum_{j=0}^{\tau_A} F_i(j)}$
  **end for**
  **return** $f$
**end function**

Provided that a detector repertoire has been built, the monitoring stage aims to determine the presenters pairing durations. To accomplish this the same decision dynamics process used on the repertoire education is used, however now with anergy (Definition 4.12). The algorithm terminates by evaluating the pairing frequency for each presenter (equation 4.1).

## 4.4 Bibliography

[1] F. Vístulo de Abreu, E. N. M. Nolte-'Hoen, C. R. Almeida, and D. M. Davis. Cellular Frustration: A New Conceptual Framework for Understanding Cell-mediated Immune Responses. In *Proceedings of the 5th International Conference on Artificial Immune Systems*, ICARIS'06, pages 37–51, Berlin, Heidelberg, 2006. Springer-Verlag.

[2] F. Vístulo de Abreu and P. Mostardinha. Maximal frustration as an immunological principle. *Journal of The Royal Society Interface*, 6(32):321–334, 2009. ISSN 1742-5689.

[3] P. Mostardinha and F. Vístulo de Abreu. Positive and negative selection, self-nonself discrimination and the roles of costimulation and anergy. *Scientific Reports*, 2:769, oct 2012. ISSN 2045-2322.

[4] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[5] A.K. Abbas and A.H. Lichtman. *Basic Immunology: Functions and Disorders of the Immune System*. Elsevier/Saunders, Philadelphia, PA, 2010. ISBN 9781416055693.

[6] Philippe Bousso. T-cell activation by dendritic cells in the lymph node: lessons from the movies. *Nature Reviews Immunology*, 8(9):675–684, sep 2008.

[7] Matthew F. Krummel, Frederic Bartumeus, and Audrey Gérard. T-cell Migration, Search Strategies and Mechanisms. *NATURE REVIEWS IMMUNOLOGY*, 2016.

CHAPTER 5

# Can the immune system perform a t-test?[1]

The self-nonself discrimination hypothesis remains a landmark concept in immunology. It proposes that tolerance breaks down in the presence of nonself antigens. In strike contrast, in statistics, occurrence of nonself elements in a sample (i.e., outliers) is not obligatory to violate the null hypothesis. Very often, what is crucial is the combination of (self) elements in a sample. The two views on how to detect a change seem challengingly different and it could seem difficult to conceive how immunological cellular interactions could trigger responses with a precision comparable to some statistical tests. Here it is shown that frustrated cellular interactions reconcile the two views within a plausible immunological setting. It is proposed that the adaptive immune system can be promptly activated either when nonself ligands are detected or self-ligands occur in abnormal combinations. In particular we show that cellular populations behaving in this way could perform location statistical tests, with performances comparable to t or KS tests, or even more general data mining tests such as support vector machines or random forests. In more general terms, this work claims that plausible immunological models should provide accurate detection mechanisms for host protection and, furthermore, that investigation on mechanisms leading to improved detection in "in silico" models can help unveil how the real immune system works.

[1] chapter submitted as: Bruno Filipe Faria, Patrícia Mostardinha, and Fernão Vístulo de Abreu. Can the Immune System Perform a t-Test? *PLOS ONE*, 12(1), jan 2017. doi: 10.1371/journal.pone.0169464

## 5.1 Background

It has long been debated whether the main function driving the adaptive immune system is related to its ability to maintain homeostasis [2, 3] or to eliminate foreign substances [4, 5]. On theoretical grounds it has been easier to build models that perform some level of self-nonself discrimination [6–8], even if it has been recognized that perfect self-nonself discrimination could be difficult to achieve [9, 10].

Instead of being focused in detecting foreignness, Niels Jerne and followers [11–13] proposed that the adaptive immune system would be concerned in maintaining homeostasis. Theoretical studies to support these ideas revolved around models of idiotypic networks, whose relevance however proved difficult to demonstrate in practice [14, 15]. After a remarkable initial growth in the 80's, these ideas were progressively abandoned afterwards [14]. This drawback does not necessarily disprove Niels Jerne main conceptual insights, which are more general than the specific model adopted to test them. According to Jerne, immune interactions should be concerned in maintaining a regulated dynamics with itself. Foreignness could not be associated to specific antigen, but rather emerge from a perturbation in the dynamics.

The absence of a self/nonself discrimination mechanism in Jerne's conceptual model was the main source of rejection for these ideas. In an attempt to reconcile the two views, Coutinho and Varela proposed a second generation of immune networks [16]. Varela and Coutinho developed extensive numerical work and showed that lymphocytes would be arranged in a giant connected component of self-reactive elements which would be responsible for maintaining homeostasis, plus a set of disconnected peripheral and nonself reactive clones, responsible for eliminating nonself invaders. Today, these directions are still explored with new formalisms, techniques and ideas [17, 18].

Unfortunately, none of the proposed models gained indisputable acceptance being still unclear which proposal provides an effective defence mechanism [19–22]. In this respect it is important to outline the efforts played by a growing community of computing oriented researchers who have been looking into the immune system for inspiration to build better computational algorithms. Indeed, if the immune system is competent in protecting the host from invaders, new computational algorithms could use similar strategies to detect deviations from normal functioning. Considerable high quality theoretical work has been done on different formulations, such as self nonself discrimination models [7, 9], idiotypic networks [17, 18], clonal expansion models [23, 24], and models following the danger hypothesis [25]. The number of applications studied have also been impressive. It could range from fault [26, 27] and intrusion [28–30] detection to mathematical optimization [31–33] and robot

path planning [34, 35]. Despite these advances, some have raised doubts on their relevance as compared to those accomplished in fields like artificial intelligence [36].

In any case, the artificial intelligence perspective has two important merits. On one side, it highlights that it is worth studying cellular processes that can encompass accurate anomaly detection, as this is likely to play a role in host protection. On the other side, it defines a research framework for testing the performance of competing theories proposing alternative mechanisms of immune protection. Indeed, our point of view is that the adaptive immune system should work as a sophisticated statistical (or data mining) detector, signalling immune responses whenever deviations from a normal state are detected. This would be analogous to the violation of the null hypothesis in statistical testing.

## The Adaptive Immune System: a Modelling Perspective

Even having three different lineages ($Th_1$, $Th_2$ and CTL), all T cells from the different subsets take part in a similar complex dynamics of cellular interactions with antigen presenting cells (APCs) (dendritic cells and macrophages) in lymph nodes. In particular, they all require stable contacts [37, 38] and two signals to become activated [39, 40]. The first signal involves interactions of T cell receptors with MHC molecules and a peptide fragment on APCs. The second signal is non-specific and typically involves CD28 and B7 molecules on T cells and APCs, respectively.

T cells play a central role in the adaptive immune system because, upon activation T cells proliferate and differentiate onto: $\mathrm{Th}_1$ cells, which migrate to sites of infection, where they activate phagocytes that captured microbes with fragments like those detected by the T cell in the lymph node; $\mathrm{Th}_2$ cells, which can undergo a similar type of complex dynamics with activated B cells, triggering their differentiation into plasma cells and antibody production; cytotoxic T lymphocytes (CTLs), which migrate to sites of infection and eliminate cells harbouring intracellular microbes (like viruses).

Given the central role played by T cells in initiating the immune response, the cellular frustration approach focused on modelling the interaction of APCs and T cells. The strategy has been that if plausible cellular mechanisms would be identified leading to an accurate triggering of immune responses, then this research could enlighten on the role of the different cells and signals exchanged, and what type of information the immune system senses and responds to. Indeed, it is unlikely that an inaccurate initiation of immune responses could lead to effective host protection.

To achieve flexible immunity, the immune system uses somatic recombination to build very diverse receptors, and therefore receptors are not transmitted to the organism's offspring. To accurately discriminate healthy from non-healthy cells and molecules, the adaptive immune system has to undergo a repertoire education (maturation) stage, which takes place in early life in the thymus, for T cells, and in bone marrow for B cells. Therefore, to establish viable immune protection mechanisms it is crucial to consider the two important stages in T cells' lifes, the education stage where T cells are selected to recognize displayed antigen and simultaneously they are prevented from reacting against host cells, and the activation stage taking place in specialized organs in the periphery (like the lymph nodes). In both cases, cells should interact following a similar dynamics, since one stage prepares the other.

The cellular frustration framework (CFF) makes an important assumption on the scale of the fundamental processes involved in the definition of the *self*. The CFF assumes that the *self* is a systemic entity and consequently, only mechanisms processing information arising from all constituents simultaneously are likely to conveniently model the detection mechanisms involved in host protection by the adaptive immune system. This has an important modelling consequence, because it assumes that studying how individual cells interact is not enough to define the *self* information. In fact, one the most important results in this article is the demonstration of how context dependent detection mechanisms can be built to capture systemic information, i.e., information on population properties.

To gain access to systemic information, the cellular frustration dynamics assumes that both, T cells and APCs, continuously monitor signals delivered by the cells they contact with and direct their immunological synapses towards the cell delivering the strongest signals. Experimentally it was already demonstrated that T cells can perform cellular decisions of this type(see [41] and in particular the supplementary video 4). Here we will assume that, given the extremely packed environments in the thymus and lymph nodes [42], cellular decisions are continuously taking place by both, APCs and T cells. The other assumption used is that only long contacts allow immunological synapses to mature and trigger effector functions [37, 41]. Hence, instead of being concerned with describing which cells interact with strong avidity, the cellular frustration approach [43] is concerned with which cells establish stable contacts (i.e., long-lived interactions). Furthermore, it will be the increase in the number of long long-lived interactions that will signal the degree of pathogenicity.

In this paper we will show that if one accepts the cellular frustration description of cellular interactions in the adaptive immune system, then the immune system should be capable of signalling the abnormal presentation of peptides with accuracies comparable to well-known statistical tests such as the t-test, or the KS test. This

result shows how cellular interactions can aggregate information distributed over many different APCs. So far, quorum sensing is the best known biological mechanism capturing global information on a system's configuration. In contrast to quorum sensing mechanisms which are sensitive to population frequencies, cellular frustration captures information of the joint frequency of several peptides in the population, and hence it aggregates more information.

## 5.2 Cellular Frustration Framework as an Unstable Matching Problem

The cellular frustration approach to the adaptive immune system received inspiration from a well-known problem in computational mathematics [44], the stable marriage problem (SMP). This problem was first proposed by Gale and Shapley in 1962 [45]. Due to its relevance to market creation, work in this area was awarded the 2012 Nobel Prize in Economics. The SMP found several applications such as in organ transplant allocation [46] or management of communication networks [47].

In the stable marriage problem, researchers look for efficient algorithms matching men and women in stable pairs [45, 48]. Finding stable solutions – i.e., arrangements with only stable pairs – can be difficult because men and women all have different and complex preferences which can interfere with each other. As a result, some instances of the problem can be NP complex [48], which means that in these cases there is no known deterministic algorithm that finds a stable solution in a reasonable computational time. In any case, even for instances for which efficient algorithms exist, they are not likely to be relevant in the context of cellular immunological interactions, since they require that cells follow a precise sequence of interactions simultaneously.

The cellular frustration approach to the adaptive immune system uses the Gale and Shapley original model as a starting point. Two cell types, T cells and APCs, play the role of men and women. The crucial difference between the two models lies on their aim. The stable marriage problem looks for stable matchings because it is argued that men and women lose time when they engage in unstable matchings. By contrast the cellular frustration framework (CFF) tries to find a subset of T cells engaging in unstable (frustrated) interactions with APCs. Since effector functions require forming stable conjugations (stable pairs), in highly frustrated populations cells are rarely activated despite of their natural tendency to interact. As a result, in CFSs cellular activation is only triggered either in physiologically tolerable numbers or when the dynamics is disrupted due to changes in the information presented by APCs to T cells. Here it will be shown that, besides responses to nonself ligands

(see [49]), T cells can be activated if a combination of self-ligands deviates from their typical frequencies of appearance. This is a type of anomaly response that does not require the presence of nonself ligands and for this reason we call it detection of abnormal-self.

As in the SMP, in cellular frustrated models it is assumed that APCs and T cells have preference lists, named interaction lists (ILists). Following [49], it is assumed that APCs discriminate only the presence of either one of two ligands on T cells, and consequently T cells can be grouped in two cell subtypes. Likewise, APCs can be grouped in two cell subtypes depending on which ligand they rank first. Therefore, in the model studied in this article, there are two cell types and two cell subtypes, as depicted in figure 5.1. For simplicity it is also assumed that all cell subtypes have $N/2$ cells.

In the CFF it is assumed that the information displayed on cells' surfaces can be mapped onto a single ligand. APCs can display a large diversity of possible ligands. On the other hand, T cells display only 2 possible ligands, which is used to define the T cell subtype. APCs subtype is also defined using these 2 ligands as it is assumed that APCs prioritize interactions with T cells of the same subtype.

The typical complex dynamics that can emerge in the SMP arises in cellular frustrated populations as well, because of the complex organization of the ligands displayed by APCs in T cells ILists. In particular, if a T cell would rank in the top position the ligand displayed by an APC of the same cell subtype, then a conjugation between the two cells would be maximally stable. By contrast, if all ligands displayed by APCs of the opposite cell subtype would be ranked first then conjugations would be short lived because even when the T cell is conjugated to the ligand ranked in the first position, all APCs of the opposite cell subtype can destabilize the conjugation. Cells destabilizing conjugations are said to frustrate interactions. The cellular frustration framework puts a special emphasis on the importance of frustration to organize the dynamics and perform accurate intrusion detection, as discussed next.

## 5.3 Ordering and Detection in Cellular Frustrated Populations

The cellular frustration framework assumes that conjugation lifetimes are reliable anomaly detection indicators that can be used to trigger effector functions. This poses a problem to the immune system, namely that of measuring conjugation lifetimes using cellular contacts. In [49] it was proposed that the immune system may have solved this problem using two combined mechanisms. Positive selection would

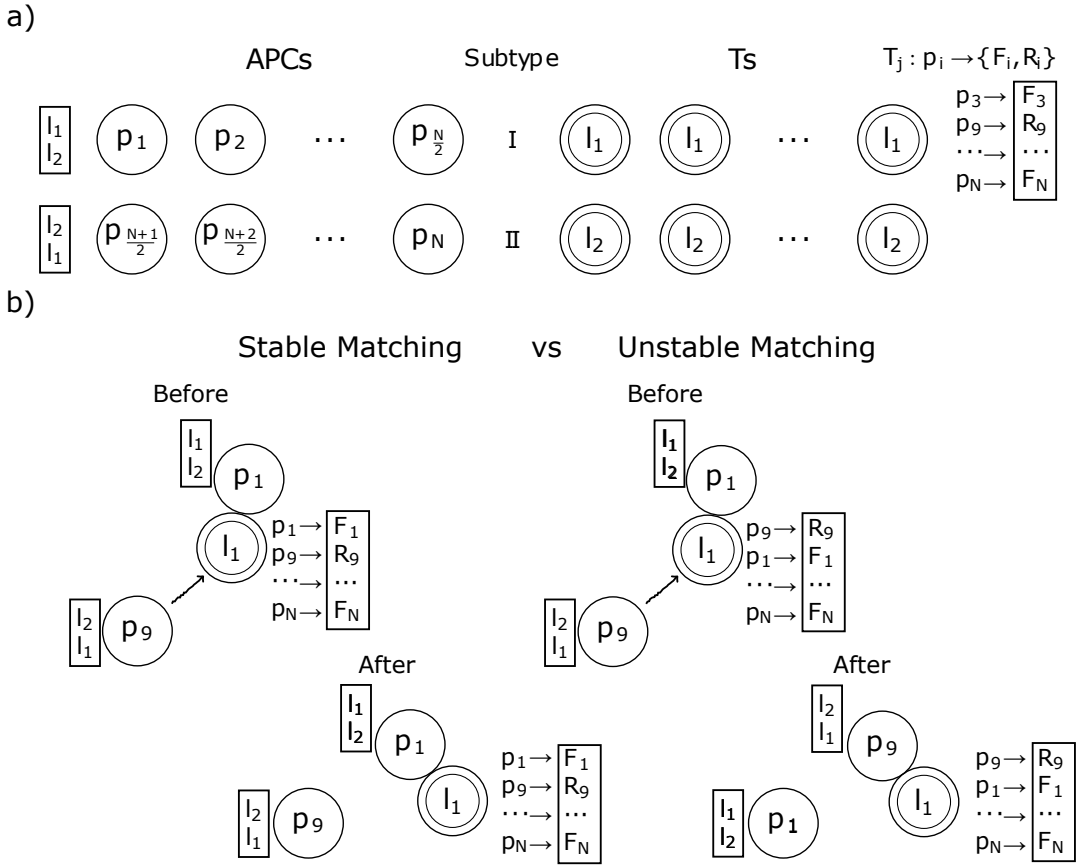


Figure 5.1: Cellular Frustration model used in this article. a) The model consists of $2N$ cells equally divided in two types, APCs and T cells. Each cell type is further divided in two subtypes, I and II, of the same size, and present ligands to cells of the opposite type. APCs present a diverse set of ligands $p_i$ ($p_i \in \mathbb{R}$) while T cells present only ligands $l_1$ or $l_2$. Ligands $l_1$ or $l_2$ determine the T cell subtype and also the APCs subtype, since APCs of subtype I (II) rank ligands $l_1$ ($l_2$) first. In this paper it is assumed that T cells map ligands $p_i$ in two ligands (or signals), one presented frequently $F_i$ and the other, $R_i$, only rarely. b) Examples of decisions taken by cells upon interaction. A new matching is formed whenever the displayed ligands are ranked higher in each other's ILists. This happens in the case on the right but not in the one on the left. In the case on the right the APC displaying the $p_q$ ligand is said to frustrate the interaction between the APC and T cell displaying ligands $p_1$ and $l_1$, respectively.

homogenize the dynamics associating to all cells conjugation lifetime distributions with close normalization factors. This allows measuring conjugation lifetimes by measuring the rate of conjugations of a given duration [50] (see Figure 5.2a).

The other important issue concerns how frustration can be changed by ordering ILists. In [50] it was shown that conjugation lifetimes are inversely proportional to the probability of destabilizing a conjugate and consequently are related to how cells prioritize interactions. In a stationary regime it was derived that conjugation

lifetimes $\tau_{ij}$ can be calculated according to:

$$\tau_{ij} \sim \frac{1}{\sum_{kp} D_{kpij} \widetilde{n}_{kp}} \tag{5.1}$$

In this expression, non-null $k$ and $i$ indices denote APCs, while non-null $p$ and $j$ indices denote T cells. Null indices are used to account for non-conjugated cells. $D_{kpij}$ is an integer that accounts for the number of ways $kp$ (a conjugate or a single cell) can destabilize $ij$. $D_{kpij}$ is 0 if $ij$ cannot be destabilized. Note that expression 5.1 is only valid in the stationary regime. Hence, $\widetilde{n}_{kp}$ is the stationary frequency of $kp$ conjugates or of $k$ or $p$ single cells (in case $k$ or $p$ are null, respectively).

Estimating the stationary frequencies $\widetilde{n}_{kp}$ can be difficult because APCs can present diverse information and also because their dynamical equations involve numerous feedbacks [50]. Therefore finding stationary $\widetilde{n}_{kp}$ frequencies demands self-consistent solutions which can only be found numerically. Nevertheless, this equation can still help us building a deeper understanding of mechanisms leading to accurate and sensitive immune discrimination.

First note that (5.1) suggests that conjugations involving cells of the same subtype are the most stable. Indeed, in that case, only the T cell may be destabilized as the APC is already interacting with a ligand that is ranked in the top position. Therefore, by considering simply the impact of non-conjugated cells in the population, equation (5.1) shows that the conjugation lifetime becomes at least inversely proportional to the sum of frequencies of non-conjugated APCs of the opposite cell type displaying ligands ranked above the ligand the T cell is interacting with. By contrast, if the same T cell is conjugated to an APC of the opposite cell subtype, the conjugate can be destabilized by all non-conjugated T cells of the opposite cell subtype, plus a number of non-conjugated APCs displaying ligands ranked higher in the IList than the ligand the T cell is interacting with. Thus we can conclude that long lived conjugations are mainly produced by conjugations involving cells of the same subtype. This is indeed what we obtain in numerical simulations.

Given the importance of distinguishing whether ligands are displayed by APCs of the same or of the opposite cell subtypes, we denote by LSCS, Ligand of the Same Cell Subtype and by LOCS, Ligands of the Opposite Cell Subtype.

Expression (5.1) also suggests that ILists could be organized to render the dynamics frustrated and allow accurate self-nonself discrimination. In [50] it was proposed that this organization could be established by the negative selection education mechanisms in the adaptive immune system. During education APCs present self-ligands. The whole set of self-ligands presented by all APCs at a given time, constitutes a configuration of the system, which changes from time to time. Negative selection eliminates all T cells interacting with high affinity with ligands

displayed by APCs and replaces them by new incoming T cells (with random ILists). Within the CFF, high affinity interactions are measured by conjugation lifetimes and consequently, T cells are eliminated because they establish stable conjugations with APCs of the same cell subtype displaying ligands that are ranked in the T cell IList top positions. Therefore and except for a small fraction, only T cells without ligands displayed by APCs of the same cell sub-type on top positions of their IList can survive negative selection. This is true for all ligands frequently displayed by APCs. However, nonself ligands were not displayed by APCs. Consequently they cannot have made any impact on the education process and therefore, T cells surviving negative selection can still have them ranked on top positions. This implies that if nonself ligands are displayed by APCs in the periphery, many T cells may establish stable conjugations with these APCs.

In [49] it was argued that the CFF offers an integrated and consistent understanding of how the immune system performs immune detection. In particular, it was shown that costimulation and anergy improve considerably the accuracy of nonself discrimination by reducing the impact of errors in negative selection. This agrees with the interpretation of experimental data [51]. The CFF proposes that anergy and costimulation guarantee that APCs are only activated if multiple different T cells – and not just a single cell which could have escaped proper negative selection – establish long conjugations with an APC. Anergy is also important to force the immune system to test a wide range of different T cell receptors.

Costimulation and anergy are two of the three signals [39] required to mount an immune reaction. Within the CFF the third signal, which is usually associated to cytokines delivered to the medium [52], naturally arises from the fact that immune reactions should only be mounted once the sum of activation signals over all cells exceeds a threshold.

Figure 5.2 summarizes the main ideas and consequences that make cellular frustration a consistent framework. In the next section we will discuss how cellular frustrated systems (CFSs) detect signals of anomalies not arising from nonself ligands.

## 5.4 Abnormal Self detection in Cellular Frustrated Systems

Consider a conjugate with cells of the same cell subtype in a frustrated population. Then, as discussed in the previous section, the conjugate can only be destabilized by non-conjugated APCs displaying LOCS that are ranked in top positions in the T cell IList. If the number of these LOCS diminishes, the denominator in equation (5.1)

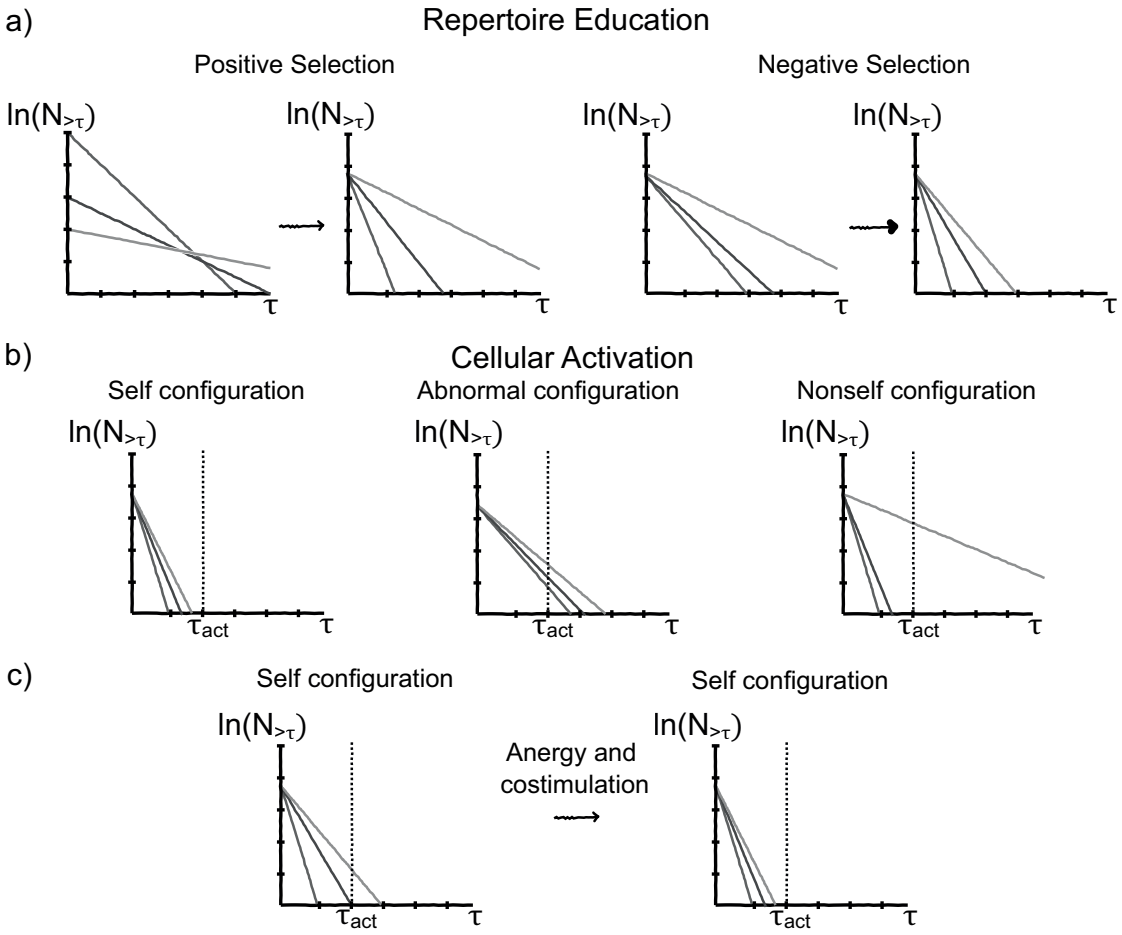


Figure 5.2: Important immunological mechanisms and their role according to the CFF. The number of conjugations lasting for a time $\tau$ is plotted in logarithmic scale on the vertical axis, for 3 representative cells in the population. a) *(left)* Positive selection homogenizes conjugation rates for all cells in the population, which is equivalent to normalizing the number of conjugations in a time interval; *(right)* Negative selection reduces the number of long conjugations. By combining positive and negative selection it becomes possible to access conjugation lifetimes - a reliable indicator given by the slope in the graphs - by measuring the rate of conjugations of a given duration. b) *(left)* Long lived conjugations, lasting longer than $\tau_{act}$, trigger cellular activation. Self-configurations are tolerated because most conjugations are short lived ($\tau < \tau_{act}$); *(middle)* in the presence of abnormal self-configurations, several cells are mildly activated; *(right)* in the presence of nonself ligands, only a few T cells interacting with nonself ligands are strongly activated. c) By using anergy and costimulation, the impact of cells that escaped education is reduced because cellular activation occurs only when several T cells produce long conjugations.

decreases, increasing the average conjugation lifetimes. Since in the CFF, long-lived conjugations trigger effector functions, this suggests that CFSs should be capable of detecting other signs of intrusion besides the presence of nonself ligands. In fact,

we will show that CFSs can detect two other types of perturbations. One is the increase in the number of ligands that are only rarely presented. These ligands are not nonself, because they appear in self-configurations. However, they appear only rarely and in this work they can be seen as the ligands appearing on the tails of distribution functions.

The other type of perturbation CFSs can react to occurs when uncommon combinations of frequent ligands are absent. This is a much more complex type of information as it captures correlations in presentation patterns. This detection mode goes well beyond single ligand statistics.

In this work it is assumed that T cells sense only two types of signals from each APC. From a modelling perspective this is equivalent to assume that each T cell maps the information displayed by each APC onto only two different ligands, i.e., as if only each APC would present one of two possible ligands. According to our results, the best immune protection is achieved when one of these ligands is perceived frequently and the other appears only rarely. This perspective agrees with that of several authors [40, 53–55]. Therefore, from here-on we refer to the information perceived as delivered by frequent or rare ligands. Note however, that this mapping differs from one T cell to the next, and that different APCs display different information and therefore their information is mapped onto different ligands. Note that different metrics can be used to establish this mapping, depending on the modelling taken to establish how T cells read the information presented by APCs. For instance, some models could focus on the interaction of the T cell receptor and the peptide-MHC complex, while others on avidity effects(see [56]).

An intuitive picture can then summarize how abnormal-self detection is achieved in CFSs. First, negative selection selects which sub-sets of ligands can be ranked in a small fraction of T cells ILists, on top positions, to minimize the frequency of long lived conjugations. Since negative selection uses a common and progressively adjusted lifetime threshold to eliminate T cells, all T cells adjust together the number of ligands kept under surveillance and in this way correlate their responses. As a result, after negative selection each T cell continuously surveys the presence of the set of frequent ligands it has ranked on top positions in its IList and displayed by APCs of the opposite cell subtype. If any of them is absent, the probability of establishing long conjugations with APCs of the same cell subtype increases. The amplitude of these individual cell responses is larger, the larger the number of absent frequent ligands. Using the long conjugation lifetimes observed in self-configurations, cellular activation thresholds can be defined.

The immune system as a whole is activated when the sum of individual cells activation signals (e.g., in the form of cytokines concentration) exceeds a threshold. This collective signalling can be related to the third signal required for the immune

system activation, and it is stronger, the larger the number of activated cells. Hence, abnormal self-detection is triggered depending on the number of frequent ligands missing and on the number of T cells having them ranked in top positions.

## 5.5 Results

Two types of results are presented next. First, the several detection mechanisms used by CFSs will be identified. This is best addressed using a specifically designed case study. Afterwards it will be shown that CFSs can achieve state of art anomaly detection performances in more general settings.

### DinBs: a case study with data presented in blocks

Consider a simplified version of the CFS model, in which $N_B$ APCs belonging to first cell subtype either present a frequent or a rare ligand. In this example, it is assumed that all APCs present different ligands and all T cells use the same criteria to establish whether APCs display a rare or a frequent ligand ($N_B = N/4$). This is a special case of the more general model considered in this article, in which all T cells perceive differently the information presented by the several APCs. Since, in the models presented here, T cells map this information onto only 2 signals (or ligands), a frequently or a rarely displayed, this is also equivalent to assume that APCs present only either a frequent or a rare ligand (Figure 5.3).

In self-configurations either $N_r^{left}$ rare ligands are displayed by the first $N_B$ APCs of the first cell subtype, or $N_r^{right}$ are displayed by the remaining APCs of the same subtype. The total number of rare ligands displayed in a configuration s is then $N_r(s) = N_r^{left} = N_r^{right} = N_r$.

Discrimination of two types of abnormal self-configurations will be tested. In the first case the number of rare ligands is increased: $N_r^{left} = N_r^{right} = N_r + \Delta N_r$. In the second case, the same number of rare ligands, $N_r$, is displayed as in self-configurations. However, now half of them are presented by cells from the last $N_B/2$ cells from the first block and the other half by cells from the first $N_B/2$ cells from the second block of the first cell subtype. A representation of this data presentation scheme is shown in Figure SM1. An extension of this model considering that rare ligands could be presented by cells belonging to both cells subtypes would lead to similar results and therefore will not be considered here.

#### Maximally frustrated populations for CFSs with DinBs

Our initial analyses consider T cell populations with partially ordered ILists. These lists maximize frustration for all cells simultaneously, minimizing the longest con-

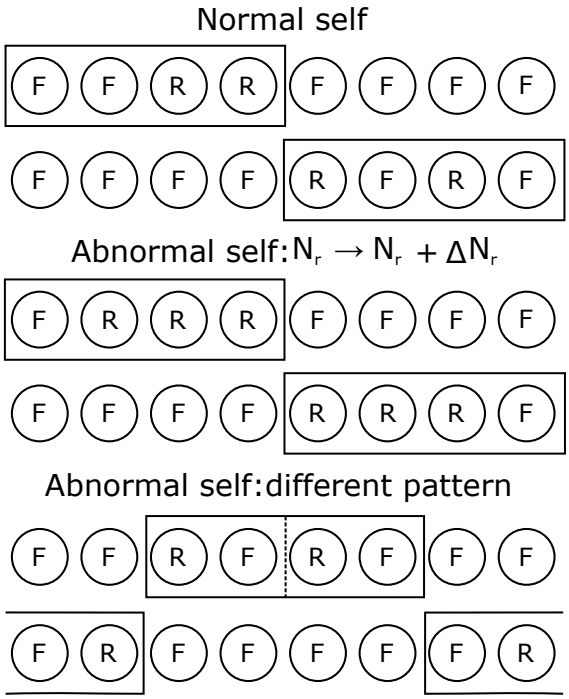


Figure 5.3: Illustration of the configurations displayed by APCs in the Data in Blocks (DinBs) case study. In circles and on each row are represented APCs of subtype I. Each APC can only present two ligands, represented either by an F or an R depending on whether they are frequently or rarely displayed. Different APCs display different ligands. Rectangular boxes delimit APCs that can randomly assigned to display rare ligands. For configurations with rectangular boxes with a dashed line, half of the number of rare ligands is displayed by APCs on one side of the box, and the other, on the other side. Configurations on the first four rows share the same presentation pattern but differ on the number of rare ligands displayed. Configurations corresponding to the last two rows, display sets of rare ligands that have never been displayed together when normal self-configurations were presented during education (maturation).

jugation lifetimes. To maximize frustration in the SCFSs with DinBs, ILists are organized according to the following strategy. First rank in top positions $N_{LOCS}^{top}$ frequent LOCSs; afterwards rank rare LSCSs; then rank frequent LSCSs; then rank rare LOCSs and finally rank the last frequent LOCSs. The specific way ligands are ranked in different T cells ILists also follows a specific order (see SM2).

### Mechanisms increasing the number of long lived conjugations

The number of stable conjugations can be increased in two ways: by increasing the probability of establishing stable conjugations by individual cells, or by increasing the number of cells establishing stable conjugations.

To establish long conjugation rates, T cells have to establish stable interactions

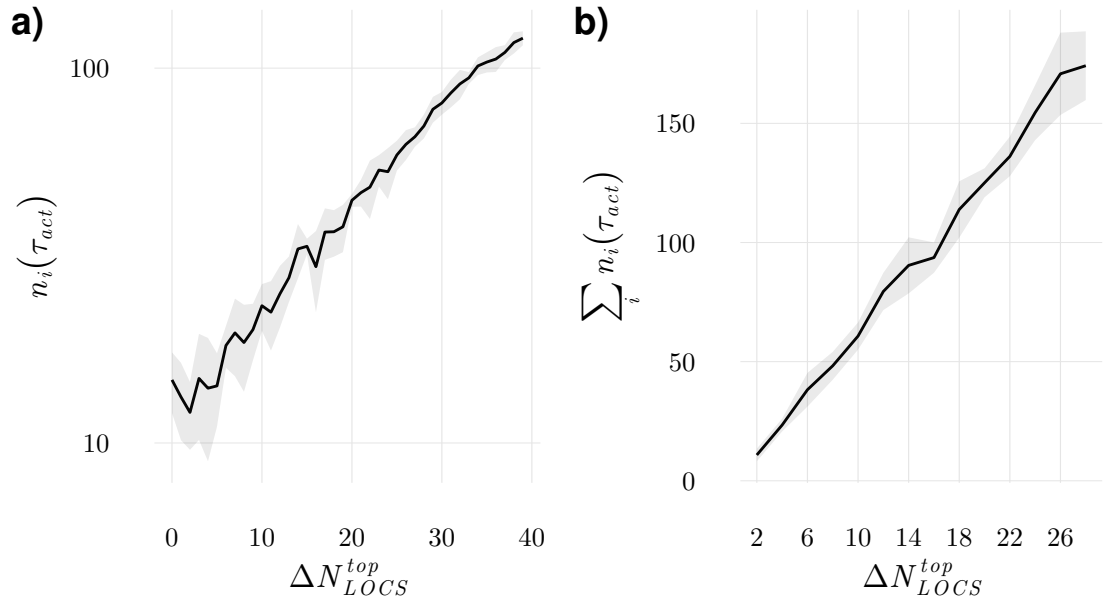


Figure 5.4: Cellular responses when ILists are specifically modified to analyse the impact of missing frequent ligands. a) the number of long lived conjugations grows exponentially when a growing number of frequent LOCS are removed from top positions in a T cell IList; b) the number of long lived conjugations grows linearly when a growing number of cells has a couple of frequent LOCS removed from their ILists top positions. In these results 100 realizations of systems with 96 cells were used.

with LSCS. This happens when frequent LOCS ranked in top positions in T cells ILists are absent. Indeed, according to equation (5.1), the largest conjugation rates increase with $\tau^{-1} \propto \sum \widetilde{n}_{k\emptyset}$, where the sum over $k$ runs over all APCs displaying LOCS on top positions in T cells ILists. One can then expect that the number of long lived conjugations established by a T cell with index $i$ should be equal to $n_i^0(\tau_{act})/n_i^0(0) = \exp(-\tau^{-1}\tau_{act}) = \exp(-\gamma N^{top}_{i,LOCS}\tau_{act})$, where $N^{top}_{i,LOCS}$ denotes the number of LOCS on top positions in the $i^{th}$ T cell IList, $\gamma$ a proportionality constant and $n_i^0(0)$ a normalization constant. If $N^{top}_{i,LOCS}$ is decreased to $N^{top}_{i,LOCS} - \Delta N^{top}_{i,LOCS}$ then $n_i(\tau_{act})/n_i(0) = n_i^0(\tau_{act})\exp(-\gamma \Delta N^{top}_{i,LOCS}\tau_{act})$. This is indeed confirmed in the results obtained in numerical experiments shown in Figure 5.4. In these experiments, all frequent LOCS are ranked on top positions in T cells ILists, except for those in one cell in which an increasing number of these ligands is moved to the lowest positions. The exponential dependency of the number of long lived conjugations on the number of removed ligands, $\Delta N^{top}_{LOCS}$, can then verified.

From these results we can conclude that T cells work as multiple correlation function evaluators responding whenever combinations of LOCS are absent. On this respect it should remarked that T cells are extremely efficient at performing this task since they evaluate multiple combinations at once. Another conclusion that

can be drawn is that the established long conjugations are specific with respect to which LOCS are absent, but are not specific relatively to which LSCSs are actually producing the long lived conjugations. Indeed, changing the ranking of LSCSs in T cells ILists does not change the probability of establishing long lived conjugations. This is in strike contrast with nonself discrimination [49] in which case the APC that triggers the response displays the ligand recognized as nonself.

Instead of requiring that T cells are heavily perturbed, it is also possible to obtain the same increase in the number of long lived conjugations by increasing the number of T cells that engage, even if only mildly, in stable conjugations in the whole population. Then, in a first approximation equation (5.1) predicts that if a number of T cell ILists is mildly perturbed, then long lived conjugation rates should increase linearly to their number. Likewise, the total number of long lived conjugations established in the whole population should increase linearly with the number of perturbed T cells. In Figure 5.4b we show that indeed, this is the case, irrespectively to which ligands disappear. This result shows that activation signals arising from multiple cells can add up to create proportionally reliable signals.

**T cells detect increments on the number of displayed rare ligands**

Now we show that T cells can detect deviations from normal presentations that are characterized by the presentation of a larger number of rare ligands than in self-configurations.

Denote by $\tau_{ij}^0(s)$ the conjugation lifetime performed by APC $i \in \mathcal{A}$ ($\mathcal{A}$ being the set of all APCs) and T cell $j \in \mathcal{T}$ ($\mathcal{T}$ being the set of all T cells) when a configuration $s \in \mathcal{S}^0$is presented ($\mathcal{S}^0$ being the set of all configurations of self-ligands presented in the thymus). Denote by $\tau_{ij}(s)$, the corresponding conjugation lifetime for configurations presented in the periphery. Then, in general, $s \in \mathcal{S}$ with $\mathcal{S}^0 \subset \mathcal{S}$.

The first important observation is that the T cell population can discriminate, in principle, the presence of any rare ligand. To see this, consider first, that in self-configurations frequent LOCS had never been absent. Then, when the rare ligand is introduced in the maximally frustrated populations defined above with all frequent LOCS on top positions in T cells ILists (and all rare LOCS on the bottom), all T cells increase their largest conjugation lifetimes (with LSCSs) since the number of frequent LOCS is reduced when a rare ligand appears. Consequently, $\max_{i\in\mathcal{A}}(\tau_{ij}(s')) > \max_{i\in\mathcal{A}}(\tau_{ij}^0(s))$, $\forall s' \in \mathcal{S}$, $\forall s \in \mathcal{S}^0$, $\forall j \in \mathcal{T}$. In principle, the same result should be obtained when only subsets of LOCS are on top positions in T cells ILists, provided all frequent LOCS appear in top positions in T cells ILists.

This was indeed confirmed in numerical simulations that counted the number of conjugations $n_j(\tau_{act})$, lasting $\tau_{act}$ instants in a fixed time interval and established

by a T cell with index $j$. In any configuration in which a rare ligand is introduced, there are always T cells for which $n_j(\tau_{act}) > \max(n_j^0(\tau_{act}))$, where $n_j^0(\tau_{act})$ is the number of long lived conjugations in normal configurations with no rare ligands displayed. Therefore discrimination is perfect in this extreme case.

The previous result requires that the number of rare ligands displayed in self-configurations is zero. This, however, is a serious drawback since it does not allow discriminating disturbances that are not linked to an increase in the number of rare ligands, but are due to different patterns of absence of frequent ligands (or, equivalently, different patterns of presentation of rare ligands).

To discriminate this type of disturbances it is crucial that frequent ligands are absent in self-configurations. In this case, however, an increase in the number of absent frequent ligands is not forcefully discriminated. This can be explained in a simple way by considering that one rare ligand is presented in self-configurations but, in the periphery, two rare ligands are presented instead. To tolerate the absence of one frequent ligand in any self-configurations, T cells should require that two LOCS ranked in top positions are absent in order to become activated. This, however, is not forcefully achieved if ILists only have subsets of frequent LOCS on top positions. In this case, triggering the system depends on the probability of activating at least one T cell. As we will show in the next section, this is the case of practical relevance, and therefore it should be studied in more detail.

Consider a maximally frustrated population with DinBs and with rare ligands presented only by APCs of one subtype in one of the two blocks. From the previous analysis, it follows that when the number of presented rare ligands is small, T cell activation should require the absence of two or more frequent LOCS ranked in top positions. The probability that $f$ frequent LOCS are absent from top positions in T cells ILists is given by:

$$P(f) = \frac{\binom{N_{LOCS}^{top}/2}{f}\binom{N_L/2 - N_{LOCS}^{top}/2}{N_r - f}}{\binom{N_L/2}{N_r}} \tag{5.2}$$

where $N_{LOCS}^{top}/2$ is the number of frequent LOCS from one block ranked in top positions, $N_r$ is the number of rare ligands presented in the configuration and $N_L/2$ is the number of ligands from one block presented by APCs of subtype 1. The probability of activating a T cell is then $P_{act} = 1 - P(0) - P(1)$. The activation of the whole system requires that a pre-defined number of T cells to be activated. This number depends on the false positive rate the system can safely accommodate. If one assumes that this false positive rate is denoted by $\alpha$, then the number of T cells that must be activated to activate the whole system, $N_a$, is the solution of the

following equation:

$$\sum_{n=0}^{N_a} \binom{N_L/2}{n} P_{act}^n (1-P_{act})^{N_L/2-n} = 1-\alpha \tag{5.3}$$

Although the number of T cells of each subtype is $N_L$, by construction only half T cells have different IList in maximally frustrated systems for DinBs. The threshold value can be found by solving the fixed point recursive equation:

$$N_{a,n+1} = N_{a,n} - \left\lfloor \lambda \sum_{n=0}^{N_a} \binom{\frac{N_L}{2}}{n} P_{act}^n (1-P_{act})^{\frac{N_L}{2}-n} + \alpha - 1 \right\rfloor \tag{5.4}$$

where $\lambda$ is a parameter chosen to guarantee convergence (typically $\lambda = 0.1$) and $\lfloor \cdot \rfloor$ denotes the floor operation. The asymptotic solution of this equation gives the (threshold) number of cells above which the system is activated as a whole: $N_a^{thres} = N_{a,\infty}$. To evaluate the discrimination power when the number of rare ligands is increased, the probability of activating the whole system is calculated from:

$$\Xi(N_r^0 + \delta N_r) = \sum_{n=N_a^{thres}}^{N_L/2} \binom{N_L/2}{n} (P_{act}^*)^n (1-P_{act}^*)^{N_L/2-n} \tag{5.5}$$

where here $P_{act}^*$ is the probability of activating a T cell when $N_r = N_r^0 + \delta N_r$, $N_r^0$ being the number of rare ligands presented in self-configurations. The results are presented in Figure 5.5, for $N_r^0 = 1, 2$ and when $\delta N_r$ is successively increased until full discrimination is achieved. Results obtained using the agent based numerical simulations with maximally frustrated populations are also depicted showing a good agreement. Furthermore, results from numerical simulations using the complete cellular frustration algorithm, with education of T cells ILists (discussed later in this article) are also shown, in Figure 5.6. The agreement among these results, confirm our interpretation of the mechanisms at play.

Two main conclusions can be drawn from these results. First, increasing successively the number of rare ligands increases the probability of discriminating the perturbation according to a logistic type of saturation growth curve. Indeed, consider the addition of a rare ligand, $N_r \to N_r + 1$. Then, in a fraction of configurations the whole system is activated. If another ligand is introduced afterwards, a fraction of the formerly non activated configurations will have a similar probability of activating the system. If one assumes that this probability is roughly constant, then full discrimination is achieved and the logistic type of saturation growth curve displayed in Figure 5.5 is explained. More importantly, this result shows that the addition of an increasing number of rare ligands can eventually be always discriminated.

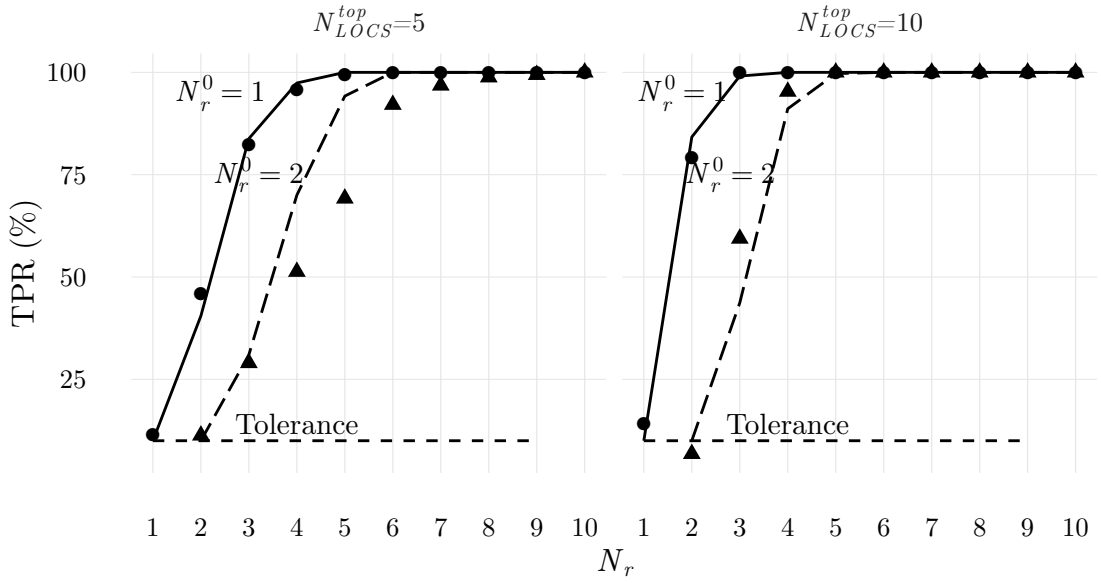


Figure 5.5: Average true positive rate when configurations with $N_r$ rare ligands are presented to populations calibrated with $N_r^0 = 1$ or $N_r^0 = 2$ rare ligands. Results in dots and triangles are from numerical simulations where T cells have perfectly ordered ILists with $N_{LOCS}^{top} = 5$ (left) or $N_{LOCS}^{top} = 10$ (right) frequent $LOCS$ in top positions from each block. Simulations used 96 cells of each type and 100 realizations. Solid and dashed lines are the predictions from the theoretical arguments described in the text. A false positive rate of 10% was assumed.

The second conclusion that is worth discussing is that the larger $N_r^0$ the harder it becomes to detect the addition of a rare ligand when $N_r^0$ is increased to $N_r^0 + 1$ ($N_r^0 \to N_r^0 + 1$). Understanding this effect is important because it highlights the trade-off between tolerance and immunity. Consider the set of all self-configurations with $N_r^0$ rare LOCS (see SM3). A large fraction of these configurations, $1 - \alpha$, has to be tolerated, while the remaining produce false positive activations.

Next consider the set of configurations with $N_r^0 + 1$ rare LOCS. To each configuration with $N_r^0$ rare ligands, $N/2 - N_r^0$ configurations ($N/2$ being the number of subtype I APCs) can be constructed with $N_r^0 + 1$ rare ligands. Clearly, the fraction $\alpha$ of activated configurations with $N_r^0$ rare ligands, will also be activated when an extra rare ligand is added. Furthermore, a set of other configurations will also become activated, totally a fraction $\delta$ of activated configurations, corresponding to the fraction of true positives. Assume that $\alpha < \delta$, i.e., there is discrimination.

Next consider what happens if the number of rare ligands in self-configurations is $N_r^0 + 1$, and the population should detect the variation $N_r^0 + 1 \to N_r^0 + 2$. From the fraction $\delta$ of activated configurations in the previous case, only a fraction $\alpha$ of the most activated configurations will activate the system, i.e., activation thresholds are

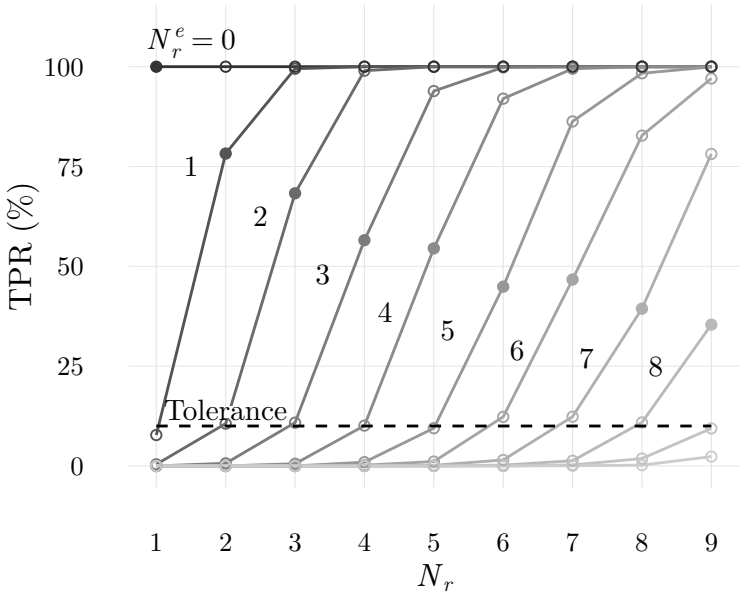


Figure 5.6: Average true positive rate when configurations with $N_r$ rare ligands are presented to populations obtained by repertoire education and calibrated with several $N_r^0$ rare ligands. Highlighted with solid dots are results obtained when a single rare ligand is added to normal configurations. Larger numbers of rare ligands in normal configurations introduce noise reducing the capacity to detect perturbations.

now more demanding. Therefore, the set of configurations with $N_r^0 + 1$ rare ligands can be divided in 3 subsets: the most activated corresponding to the false positives when $N_r^0 + 1 \to N_r^0 + 2$; the least activated, corresponding to false negatives when $N_r^0 \to N_r^0 + 1$ (configurations not activated and with $N_r^0 + 1$ rare ligands); and the remaining activated configurations when $N_r^0 \to N_r^0 + 1$. From this analysis it becomes clear that the discrimination power decreases when $N_r^0 + 1 \to N_r^0 + 2$ relatively to $N_r^0 \to N_r^0 + 1$, because the probability of creating activated configurations with $N_r^0 + 2$ rare ligands from the least activated configurations with $N_r^0 + 1$ rare ligands is smaller than the probability of generating these configurations with the intermediately activated configurations with $N_r^0 + 1$ rare ligands. Therefore, very general arguments explain the behaviour observed in the results in Figure 5.6.

In general, those configurations that previously activated the system, are now more likely to trigger the immune system when $N_r^0 + 2$ rare LOCS are displayed. This is because these configurations already have a larger number of activated cells, and therefore the number of T cells that should also become activated is smaller, or because there is a higher probability of further removing a frequent LOCS from the top of an activated T cell IList. However now, since thresholds are more demanding, only for a fraction of configurations the removed frequent LOCS can activate the

immune system and consequently it should be expected a smaller discrimination power.

In simpler terms it could be stated that introduction of rare LOCS introduces noise which makes discrimination harder. However, in the next section it will be shown that this allows detection of other types of perturbations.

The previous mathematical approach assumed that populations displayed a small number of rare ligands in the thymus. In this limit, activation was controlled by the number of T cells sensing the absence of 2 frequent LOCS in tops positions. For self-configurations with a larger number of absent frequent LOCS, the threshold controlling the systems activation may impose that a combination of T cells sense the absence of a different number of frequent LOCS (e.g., a fraction sense the absence of 2 while another fraction the absence of 3 frequent LOCS) since the systems response aggregates the responses of all individual cells. The CFF proposes a set of immunologically plausible mechanisms to make this adaptive selection of criteria automatic. Clearly, the immune system, and agent based simulations in particular, have the advantage of being able to tune thresholds to adapt to complex presentation patterns.

### T cells can sense contextual information

A more challenging detection capability consists in detecting disturbances that are not due to an increase in the number of rare ligands. Instead, it is the combination of ligands that determines whether a configuration is normal or abnormal. In this case the information is contextual. Within the CFF, T cells can still respond to this type of disturbances since responses depend on the combinations of absent frequent LOCS. However, a number of conditions have to be met. Firstly, the mapping into rare and frequent ligands should make rare ligands sufficiently frequent in order to provide information of presentation patterns. Secondly, frequent LOCS should not all be ranked on top positions. If this would be the case, then the number of absent frequent LOCS would not change as their overall number does not necessarily increase. Finally, and for the same reason, the systems response should only receive contributions from cells sensing the absence of frequent LOCS as if all cells contributed their responses could cancel out. This naturally shows why single cell activation thresholds are needed.

Now we apply the previous mathematical analysis on maximally frustrated populations in the DinBs case study. To guarantee T cell's tolerance towards the two types of normal configurations, T cells ILists should have on top positions the same number of frequent LOCS displayed by APCs of the two blocks. In this way, it is guaranteed that any T cell has at least $N_{LOCS}^{top}/2$ frequent LOCS on top positions.

This number increases to $N_{LOCS}^{top} - N_R$ if $N_r < N_{LOCS}^{top}/2$.

Consider a specific example where $N_{LOCS}^{top} = N_L/4$, i.e., each T cell has on top positions half of the whole set of frequent LOCS of each block of data. In abnormal configurations, the same number of rare ligands is presented, but now half is presented on a block on the right and the other half on a block on the left (see Figure 5.3). The minimum number of frequent LOCS in ILists top positions is $N_{LOCS}^{top} - N_R$, but now this value has 0 as lower bound see SM4. Therefore, some cells sense the absence of a considerable larger number of frequent LOCS and will respond to the contextual change.

Consider that only two rare ligands are presented, both in normal and abnormal configurations. In this case, all steps taken before to find the activation threshold, $N_a^{thres}$ are still valid since normal configurations are the same. What differs is the population response to abnormal configurations. Since T cells should not have all frequent LOCS on top positions in their ILists, they sense the presence of frequent LOCS differently.

By construction, when $N_{LOCS}^{top} = N_L/4$, there are 2 ILists (identical) with $N_c$ frequent LOCS in the central block (half on the left and the other half on the right) and $N_{LOCS}^{top}/2 - N_c$ frequent LOCS outside. Since one rare LOCS is presented on each side, the probability that 1 frequent LOCS is missing from top positions in an IList is:

$$p_1 = \frac{\binom{N_c/2}{1}\binom{N_L/4 - N_c/2}{0}}{\binom{N_L/4}{1}} = 2\frac{N_c}{N_L} \tag{5.6}$$

The probability that a T cell with $N_c$ frequent LOCS in the central block, misses 2 frequent LOCS (one on each side) is:

$$P(2, N_c) = p_1^2 = 4\frac{N_c^2}{N_L^2} \tag{5.7}$$

Finding the probability of having more than $N_a^{thres}$ cells with 2 frequent LOCS missing is harder to calculate now than in the previous case. It amounts to calculate a convolution with $N_a^{thres}$ probability distributions. We computed this numerically by using a Monte Carlo approach. For a system with 48 APCs of each subtype ($N_L = N/2 = 48$) and when $N_{LOCS}^{top} = 24$, we obtained $N_a^{thres} = 9$ and $\Xi_{context}(N_r = 2) = 13.5\%$ for $\alpha = 0.1$.

This agrees with the average true positive rate of 14.1% obtained after simulating 100 populations using the cellular frustration dynamics on populations with maximally frustrated ILists and shown in Figure 5.7. There, detection rates obtained when the number of frequent LOCS in top positions is varied are also shown.

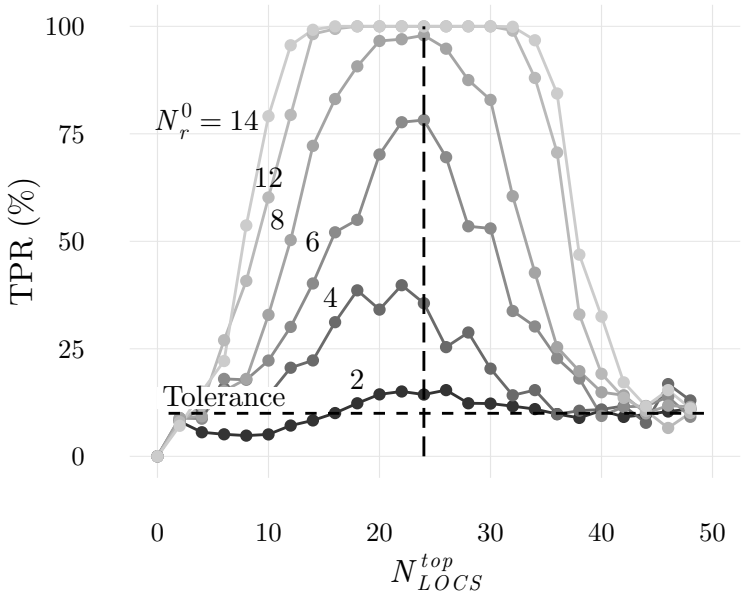


Figure 5.7: Average true positive rate for simulations for context dependent (abnormal self) discrimination. In the simulations considered for these results, both normal and abnormal configurations displayed the same number of rare ligands, although in different patterns as shown in Figure 5.3. Averages accounted 100 realizations and a false positive rate of 10%. Populations with 96 T cells and partially ordered ILists with $N_{LOCS}^{top}$ were used. From these results it is clear that the best discrimination is achieved for partially ordered ILists that maximize the number of (potentially) absent frequent LOCS in top positions. In the present case this number is 24, corresponding to the size of an ILists with all LOCS from the block presented in the abnormal self configuration. It is also clear that the larger the number of rare ligands displayed, or equivalently, the larger the number of absent frequent LOCS, the higher is the discrimination.

Two important conclusions can be drawn from these results. Firstly, the number of frequent LOCS that should be ranked in top positions to maximize discrimination depend on the data presented. In this case rare ligands were presented in blocks with 24 APCs of the first subtype, and the maximum number of different ligands between normal and abnormal configurations was 24. As a result, the number of frequent LOCS that should be ranked in top positions to maximize discrimination was also 24. This shows that the immune system should find ways to automatically adapt to the information presented, ranking only an adequate number of frequent LOCS in top positions. This, of course contrasts with what was required to obtain the best results if the immune system was only concerned with detecting an increase in the number of displayed rare ligands. Therefore a trade-off exists as highlighted by these results.

Secondly, even if there are 24 frequent LOCS ranked in top positions (results

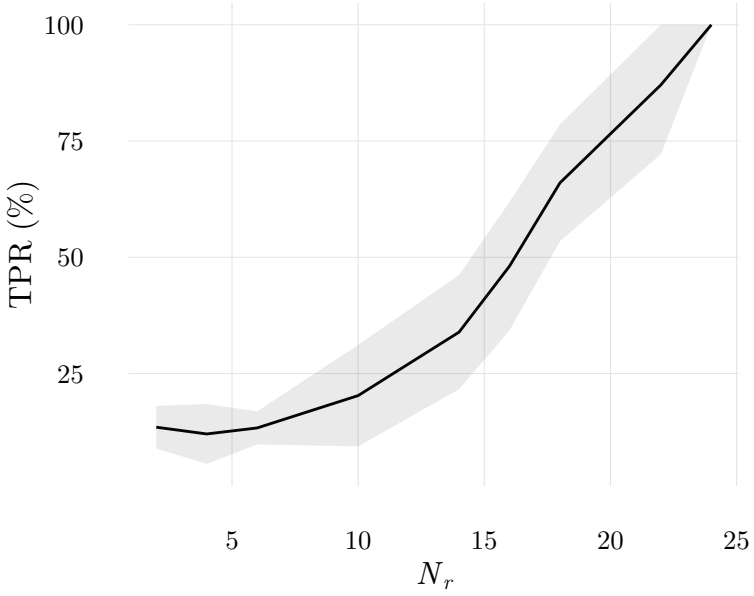


Figure 5.8: Abnormal self discrimination for populations with educated ILists when the number of rare ligands displayed is varied. Increasing the number of rare ligands in self and abnormal self-configurations increases discrimination. This result agrees with the theoretical arguments developed under the more restrictive conditions of Figure 5.7. In particular, it shows that context information can be perfectly discriminated even when ILists are ordered by negative selection.

along the dashed vertical line in Figure 5.7, the number of absent ligands is important to achieve the highest discrimination. In Figure 5.7, a minimum number, higher than 16, is required to obtain a 100% true positive rate. Increasing beyond this number, does not improve discrimination of contextual information, but it could be prejudicial for discriminating increments in the number of displayed rare ligands.

In Figure 5.8 the true positive rate obtained with populations of T cells educated following negative selection, are also shown. These results show that the mechanisms of negative selection, which will be analysed more thoroughly next, also demonstrate that discrimination of contextual information is possible and improves with the number of rare ligands displayed. Results obtained with negatively selected populations lead however to poorer performances, which is expectable since maximally frustrated populations had been designed to obtain the best performances.

### Negative selection is heterogeneous and selects T cells with the largest number of LOCS on top positions

The previous analysis provided two important results. The first was that contextual discrimination is possible and, in certain cases, even perfect. The second was that ILists should have on top positions a balanced number of ligands of each

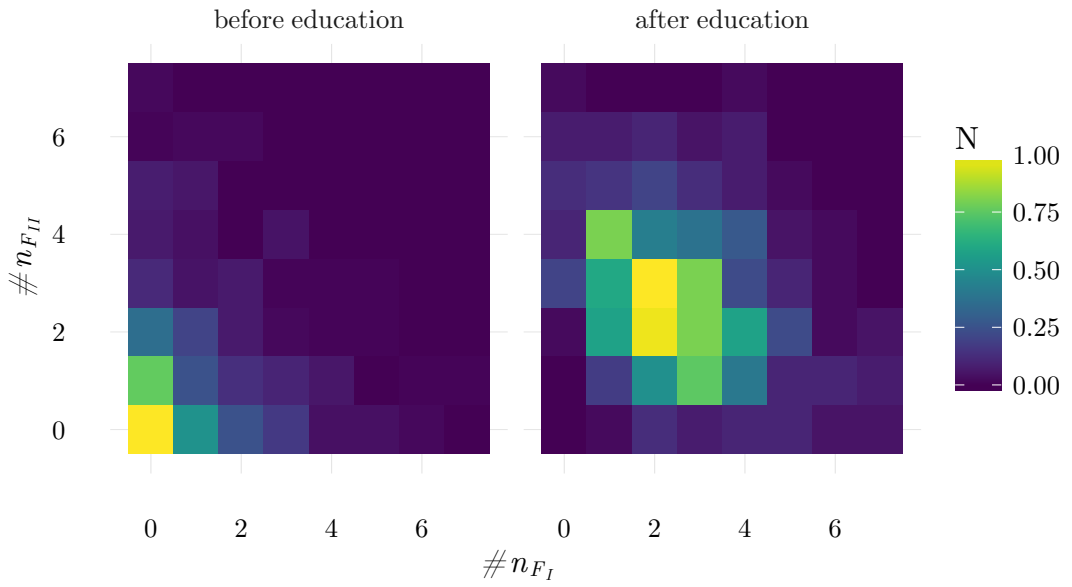


Figure 5.9: Frequency of the number of frequent LOCS of each block (I and II) on ILists above the highest ranked LSCS, before and after education (maturation). These results were obtained considering self-configurations with $N_r = 6$ (see Methods for other remaining simulation parameters). These results show that repertoire education balances the number of frequent LOCS of each block on ILists top positions, guaranteeing that a minimum number of frequent LOCS is always present in self-configurations.

block so that there is always a minimum number of frequent LOCS present in any self-configuration. The important issue that should be addressed next consists in understanding how ILists organization and the number of LOCS in top positions can be chosen autonomously. In the adaptive immune system this is achieved in the thymus through repertoire education mechanisms. Within the CFF, negative selection operates by eliminating T cells producing the longest conjugation lifetimes. Since the longest conjugations occur for those cells sensing the absence of the largest number of frequent LOCS, negative selection should select T cells that rarely have less than a (threshold) number of frequent LOCS absent. When applied to the DinBs case study this is indeed what is observed, as shown in Figure 5.9. Therefore, negative selection produces the ordering required for contextual discrimination, as discussed in the previous section.

Figure 5.10 shows two other important points. Firstly, when the maximum conjugation lifetime decreases, the average position of the LSCS ranked in the highest position in T cell ILists lowers, i.e., more LOCS are ranked in top positions. Secondly, the T cell population is considerably heterogeneous with the number of LOCS in top positions ranging from less than 5 to up to 10. This happens in spite

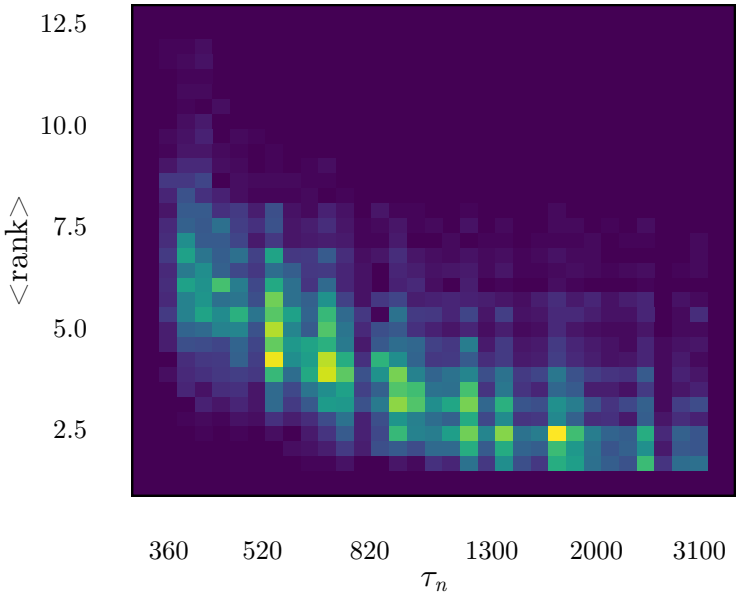


Figure 5.10: Two dimensional histogram for the mean rank of the first ranked LSCS in each IList as a function of the maximum conjugation lifetime $\tau_n$ in the population along repertoire education. It is clear from these results that LSCSs occupy progressively positions more on the bottom of ILists. For this figure 10 repetitions of the education of 120 populations with 96 T cells were considered. Self-configurations had $N_r = 6$.

of the fact that all T cells are equivalent in terms of their ILists organization. In this case, however, stochastic fluctuations are enough to break down this idealized equivalence. In practical applications, other sources of variation exist since not all APCs present the same information and also because, when limited connectivity is imposed on T cells ILists, T cells do not interact with all APCs.

### Negative selection pushes frequent and rare LOCS at different rates

The adaptive nature of T cell repertoire education can also be appreciated by noticing that frequent and rare LSCS tend to occupy different positions in T cells ILists after repertoire education. In particular, the highest ranked rare LSCS tend to be ranked higher than the highest rank frequent LSCS. This happens because negative selection operates more frequently on frequent LSCSs than on rare LSCSs. The paradigmatic example happens in the extreme case in which rare LSCSs do not appear during education, in which case they are nonself. Then negative selection cannot have an impact on how they are ranked in ILists. This can be confirmed in Figure 5.11, which shows that nonself ligands are ranked uniformly, while frequent LSCS have higher probability of occupying positions away from the top. Rare ligands appear less frequently then frequent LSCSs and consequently they have

higher probability of accumulating in higher positions in ILists than frequent LSCSs.

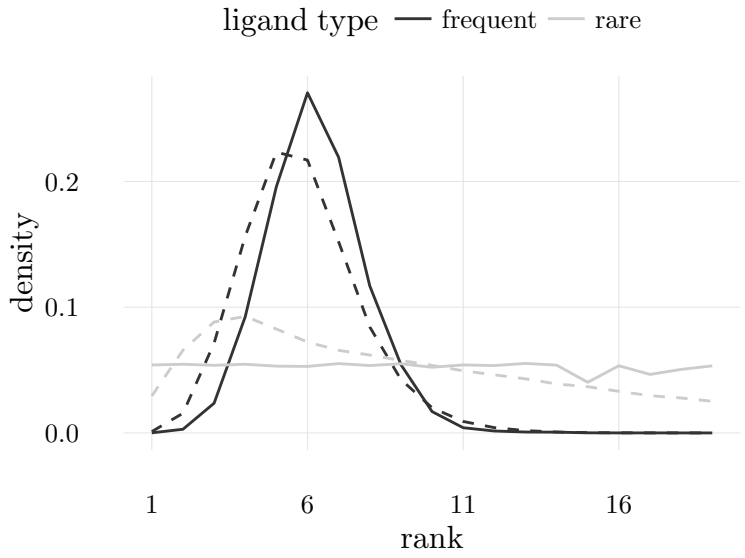


Figure 5.11: Distribution of the position occupied by frequent and rare LSCSs (dark and light coloured lines) after repertoire education for configurations with $N_r = 0$ (solid lines) and $N_r = 6$ (dashed lines). It is clear from this figure that frequent LSCSs tend to occupy lower positions than rare ligands. When $N_r = 0$, rare ligands are uniformly distributed in ILists, since they do not appear during repertoire education. When $N_r = 6$ the set of ligands appearing during education is larger making is more difficult to order ILists. As a result, in this case frequent LSCSs tend to occupy higher positions in ILists than frequent LSCSs in the $N_r = 0$ case.

Another important observation concerns the impact of the number of ligands presented on the ordering of ILists after negative selection. The distributions of rare and frequent LSCSs are centred on higher positions than the distribution of frequent LSCSs when nonself ligands are displayed. In fact, in the first case, the number of ligands that have to be "educated" is higher than when nonself ligands are presented, in which case there is effectively only half of the number of ligands.

There are two sources of limitations that prevent reaching perfectly ordered ILists. The first is statistical: the probability of randomly drawing a perfectly ordered IList decreases exponentially with the list size. The second originates from the intricate frustrated dynamics. When a T cell establishes a long conjugation with an APC, it prevents other T cells from interacting with the same ligand. As a result some of these T cells can be eliminated by negative selection since they sense a higher number of absent LOCS to this indirect effect. Every time a T cell is eliminated by negative selection, it is replaced by another nave cell with a random IList, and consequently, in the whole population there is a high probability that it

may be eliminated again in another upcoming configuration. Furthermore, every time a T cell establishes stable conjugations with some LSCS it may send other T cells to education as a side effect, making it hard to reach a state where all T cells have a similar probability of maintaining tolerance.

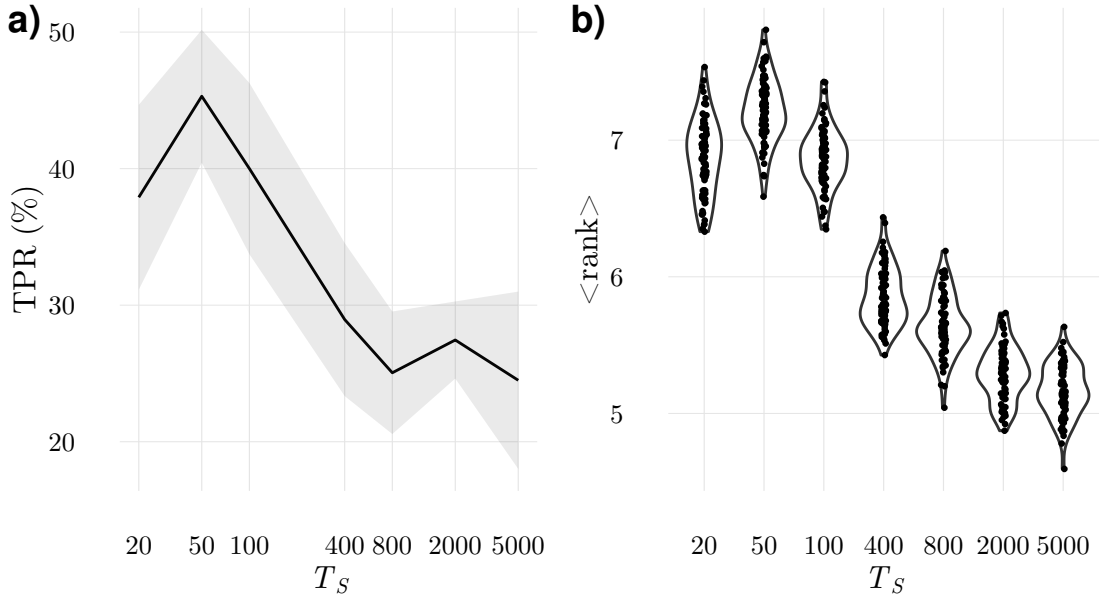


Figure 5.12: Generalisation capabilities are gained when samples are changed after every short time interval, $T_S$. a) true positive rate for discrimination of configurations with an added rare ligand ($N_r = 7$) b) mean rank of the highest ranked LSCS in T cell ILists in the 200 samples used for education. These results show that ordering of ILists is best achieved when samples are changed every $T_s = 50$ iterations which also leads to the best discrimination. Since $\tau_n$ - the maximum conjugation duration used to eliminate T cells by negative selection - is changed only when no cell is eliminated in the last $W_\tau = 10000$ iterations, this forces ILists to be consistent with the last $10000/50 = 200$ samples.

### Generalisation capabilities depend on the presentation frequency

Another factor contributing to the creation of fluctuations during education is the rate of change of the information presented by APCs (presentation frequency). In our simulations, the configuration of ligands presented by APCs changes every $T_S$ iterations. $T_S$ has an important effect in the ordering of ILists and in detection. For very large values of $T_S$, education strictly avoids having LSCS on top positions in any IList. For small values of $T_S$, to be eliminated during education, it is required that in two consecutive configurations a T cell lacks frequent LOCS on top positions in its IList. Therefore, LSCSs, and in particular, rare LSCSs can appear on top positions in some ILists because the probability that they appear in consecutive

configurations is $p_R^2$ which is small ($p_R^2 \sim 2\%$ when $p_R = 15\%$). Consequently, when $T_S$ is small there will be more LOCS in top positions, on average, even if there may be some rare LSCSs among them. This agrees with the sequence of results in figures 5.12).

## Applications

In the previous sections it was shown that CFSs can perform elaborate discrimination tasks. Now we evaluate how these discrimination capabilities compare to the performance of well-known statistical tests. We end by considering even more complex scenarios for which one must resort to data mining algorithms.

### Can the immune system perform a t-test?

Probably the best known and most widely used statistical test is the t-test. For this reason we questioned whether cellular systems could perform this test with comparable accuracy. To define self-configurations, ordered samples with 80 numbers were randomly drawn from a Gaussian distribution with an average of $\mu_S = 50$ and standard deviation $\sigma = 10$. Abnormal self-configurations were obtained by drawing samples from Gaussian distributions with the same $\sigma$, but with $\mu_{NS} = 50 \pm \Delta$ (two sided tests). The $i^{th}$ APC displayed the $i^{th}$ number in the sample as a ligand. Therefore, the first APC always presented the smallest number while the last APC, the highest. T cells sense these ligands as R or F ligands, depending on whether the number falls inside or outside an interval where a fraction $v_j$ of the numbers displayed by an APC in normal configurations lie. Here, $j$ is the T cell index, and $v_j < 50\%$ since it corresponds to the frequency of rare ligands.

In this article, T cells define rare ligands intervals for all ligands either on the right or on the left tail. Their size is controlled by the discrimination parameter $v_j$ which is drawn from a uniform distribution between 0 and $v_{max}$. As a consequence, T cells with $v_j \approx 0$ are only sensitive to nonself ligands.

To assess how CFSs detect changes in $\mu_S$, the T cell population underwent negative selection, during which 1000 self-configurations were presented by APCs. Afterwards, APCs presented another 1000 self-configurations and 1000 abnormal self-configurations. In Figure 5.13, receiver operating characteristic curves (ROC curves) are presented for different changes in $\mu$, with $\Delta = \pm 1, \pm 2, \pm 3$. Comparison with the results that would be obtained with a t-test or a KS-test are also provided.

The best results are obtained with a t-test, as expected because data was prepared respecting its assumptions, and also because the t-test is a uniformly most powerful test for detecting deviations on the distribution mean [57–60]. By contrast and as expectable, the KS-test has the lowest performance since it is a simple

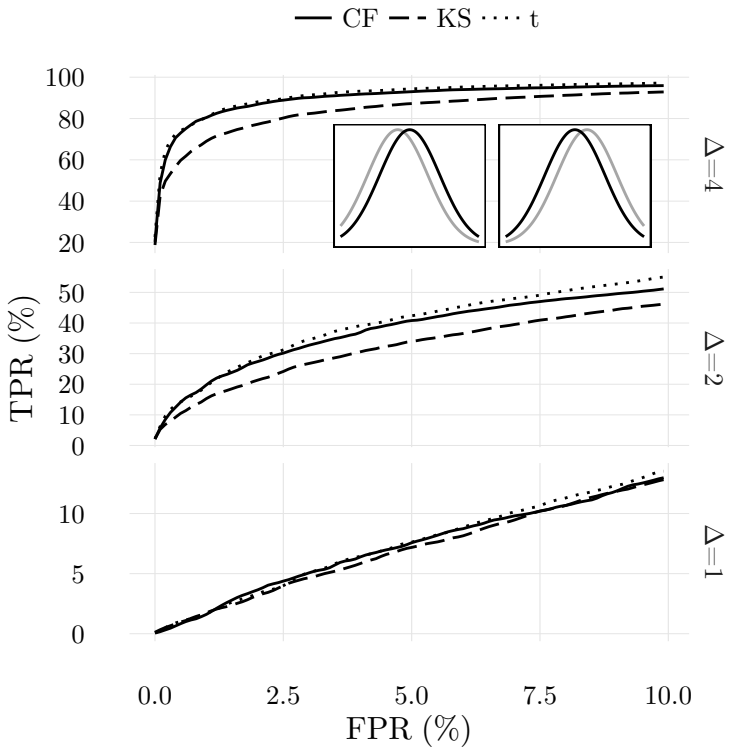


Figure 5.13: Comparative analysis of average ROC curves obtained in location tests using ordered samples with 80 elements drawn from normal distributions for the cellular frustration model and two-sided KS-test and t-test. Insets: comparison between gaussian distributions used to draw self-configurations (black) and abnormal configurations with deviations to either side (gray). Similar plots would be obtained for the distributions in the bottom examples. Abnormal-distributions are only slightly displaced. The t-test is the best estimator as demonstrated in the literature for this ideal case. However, cellular frustration models give very close results.

nonparametric test. Results obtained with CFSs are, to a certain extent surprising since little assumptions were necessary on the type of information displayed by APCs and on the type of changes that could take place.

The CFF took three important assumptions in this example. The first was that the several elements in a sample could be ordered since all derived from a same Gaussian distribution. The same assumption is used in the KS test. The second assumption consisted in placing rare ligands in the tails. This guarantees that if the distribution deviates to either side the number of rare ligands present in the sample increases, and therefore this change has a high probability of being detected. Finally, the third assumption consisted in assuming that rare ligands in all APCs tend to appear on the same side of the distribution. This assumption makes sense because, given the ordering applied to the elements in the sample, a deviation in

one tends to produce similar deviations in the next.

The three assumptions taken are important to obtain the best results, and they show if the immune system uses a cellular frustration strategy to perform discrimination, then evolution could have played a role to incorporate similar restrictions that allow reducing the space of potential disturbances. In the present problem, the first assumption makes sense given the nature of information presented. The second and third assumptions, could be relaxed if it would be important to contemplate a wider range of challenges. The same type of trade-offs distinguish parametric and nonparametric tests in statistics. In this sense, CFSs work as nonparametric tests, and are more accurate than KS tests, although computationally more involved. In fact, an interesting link exists between the KS approach and the CFF. In the KS test, the maximal deviation from a standard cumulative distribution determines whether the null-hypothesis should be abandoned. This decision crucially depends on sampling ordering. In the CFF the same deviations appear in the form of missed frequent ligands. The advantage of CFSs is to account to all deviations and furthermore to the presence of nonself ligands.

Another issue that should be addressed consists in knowing whether CFSs are robust if non-optimal parameters, such as $v_{max}$, are used. This is particularly important because the optimal parameters typically depend on the challenge. In Figure 5.14 we show how results for a false positive rate of 5% change when $v_{max}$ is changed. Clearly, when $v_{max}$ is in the range $[0.05, 0.15]$, results are consistently good for the different perturbations. This result is interesting because it shows that the best results are obtained when a fraction of cells perform abnormal-self detection, and not simply the detection of nonself ligands, i.e., outliers. From these results, we can conclude that the CFF contains the necessary mechanisms to tackle discrimination tasks conceptually equivalent to those solved by t-tests, an interesting result given the differences in the methods.

**Can the immune system perform robust statistics?**

An important area of research in statistics is concerned with improving the performance of statistical methods when the conditions required for their application are not met. CFSs are computationally more involved, so one could question if they offer robust solutions. We use the simple and paradigmatic example of the performance of t-tests in location tests when samples derive from a log-normal distribution. This case is challenging to the t-test because, even though, according to the central limit theorem, the distribution of the mean should converge to the normal distribution, the presence of heavy tails makes this convergence slow. This is particularly true for small samples. Consequently, the t-test can fail considerably.

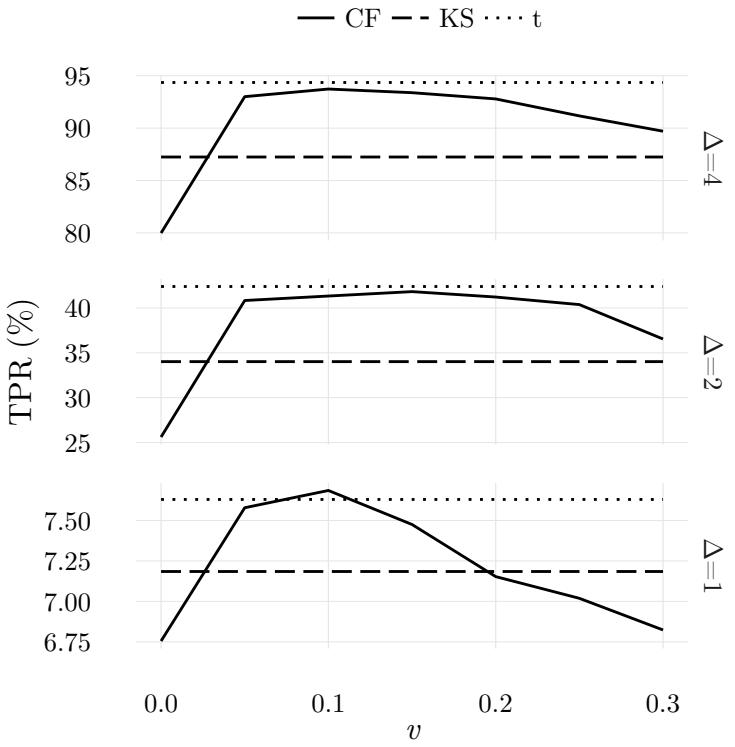


Figure 5.14: Average true positive rates obtained when the maximal probability of perceiving the information displayed by an APC as a rare signal (or ligand) is $v_{max}$ for the three location tests conducted. Discrimination is best for $v_{max} > 0$ which means that discrimination is not exclusively due to the presence of nonself ligands. A 5% false positive rate was considered.

To test the accuracy of CFSs in this case, 1000 ordered samples with 80 numbers were drawn from log normal distributions with the same means and standard deviations as in the previous example. Furthermore, $v_{max} = 0.05$ was used and the same mapping into frequent and rare ligands was chosen. The ROC curves obtained are shown in Figure 5.15. CFSs demonstrate robustness over the range of perturbations considered.

**Is the immune system a sophisticated data miner?**

In more realistic scenarios, many pieces of information have to be accounted before triggering an immune response. Each of them can depend on different sources of variation and consequently a multivariate analysis is required. A simple way to mimic this more general scenario is to assume that in the previous examples samples are not ordered, i.e. each element in a sample is drawn from independent generators. Consider next the example in which, in self-configurations, each APC presents a ligand drawn from a Gaussian distribution centred at $\mu_S = 50$ and with

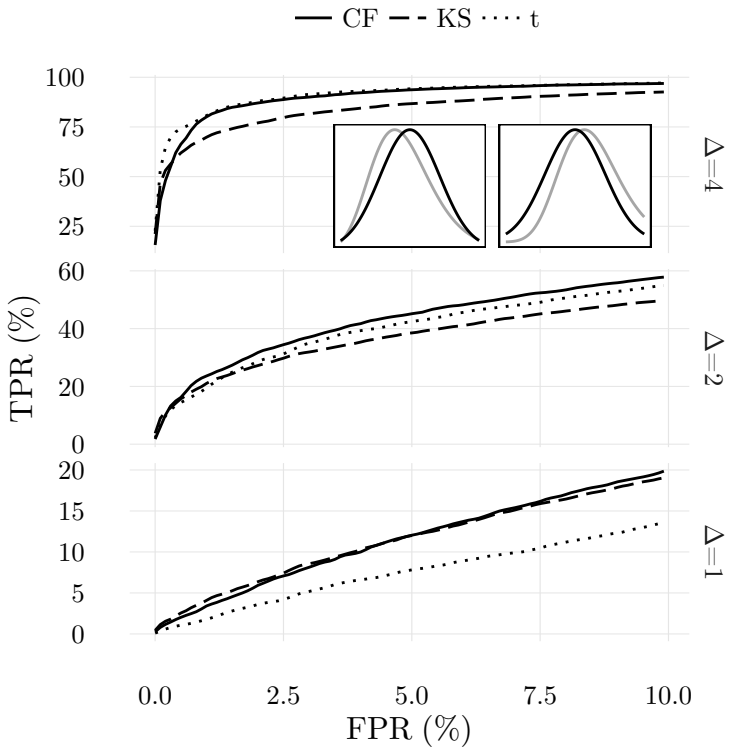


Figure 5.15: Comparative analysis of average ROC curves obtained in location tests using ordered samples with 80 elements drawn from lognormal distributions for the cellular frustration model and two-sided KS-test and t-test. Insets: comparison between the lognormal distributions used to draw self-configurations (black) and abnormal configurations with deviations to either side (gray). Cellular frustration models outperform the other statistical tests.

standard deviation $\sigma = 10$. Furthermore, in abnormal configurations $\mu_S$ is changed to $\mu_{NS} = 50 \pm 4$. These changes occur independently for each APC and in each configuration.

To evaluate CFSs responses, the same strategy used to map APCs ligand information into rare or frequent ligands and to select the T cell repertoire is taken, and ROC curves calculated. To evaluate the performance of CFSs, a comparison with the best data mining techniques for this type of tasks was established (see Methods for more implementation details of these methods). The results are presented in Figure 5.16.

One-class SVMs, just like the CFSs defined here, only use self-configurations to infer what is normal and abnormal. Therefore, CFSs and one-class SVMs are methods that use the same information to accomplish the same task. The results shown in Figure 5.16 show that CFSs perform considerably well, outperforming one-class SVMs. Note nevertheless, that our goal here is not to argue that this may

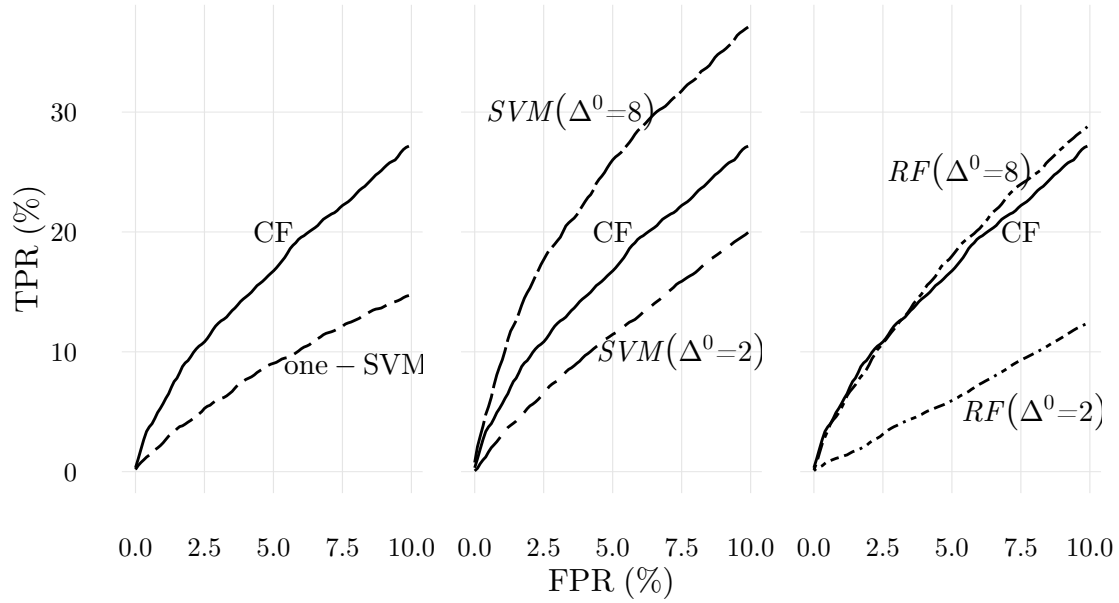


Figure 5.16: Comparison of the performance of the cellular frustration (CF) model with state of the art data mining algorithms. On the left the results obtained from CF model are compared with one-class SVM. The CF model clearly outperforms the data mining algorithm. In order to confirm that the results from the CF model are realistic, the same results are compared with support vector machines (center) and random forests (right) classification algorithms. Classification algorithms need to be trained with samples from abnormal configurations (drawn from Gaussians with mean deviated by $\Delta^0$), and consequently the comparison with the CF model, which uses only information from self-configurations, would be unfair. However, it shows that even with extra information, classification algorithms can fail to produce good results. This happens when the examples of abnormal configurations are not sufficiently distinct from those belonging to self-configurations. See text and Methods for more details.

always be the case. In fact, SVMs solve an optimization problem and consequently datasets could be designed for which they provide the best solutions. Our goal is to demonstrate that it is not hard to find relevant problems for which CFSs are better, at least when standard approaches are used (i.e., using standard sets of parameters see Methods).

Since one-class SVMs performances were so poor in comparison to CFSs, another set of numerical simulations was conducted to compare CFSs with classification methods, like two class SVMs and random forests (RFs). Classification methods require both, normal and abnormal configurations for training. Therefore, a direct comparison with CFSs is not straightforward. However, classification methods can indicate whether the information captured by CFSs and depicted in Figure 5.16 is indeed present, and how classification methods behave when the information provided for training does not perfectly match the information required for detection.

In Figure 5.16 a comparison of the performance of two class SVMs and random forests (RFs) with CFSs is presented. In these results, classification methods were trained with two sets of samples of the same size and generated from Gaussian distributions centred at $\mu = 50 \pm \Delta^0$. For self-configurations, $\Delta^0 = 0$, while for abnormal self-configurations $\Delta^0 = 2$ or $\Delta^0 = 8$. For detection, configurations centred around $\mu = 50 \pm \Delta$, with $\Delta = 0$ and $\Delta = 4$, were used. These results show that if classification methods have access to abnormal configurations that do not strikingly contrast with self-configurations ($\Delta^0 = 2$), then discrimination of abnormal configurations centred around more distant positions ($\Delta^0 = 4$) can be problematic. In brief, classification methods work well only if distinctive examples are consistently presented. This, of course, can be problematic if only mildly abnormal self-configurations are available. In more mundane terms one could argue that classification methods are only good at predicting the occurrence of earthquakes if a strong earthquake of the same type has been felt before. Finally, these results also show that the immune system can still improve its performance if it learns from examples of abnormal self-configurations. This, we know, the immune does, through clonal expansion mechanisms, a topic that we leave for a future publication.

## 5.6 Conclusions and Perspectives

The results reported in this paper represent a landmark in the cellular frustration framework and its relevance to immunology because they clearly show that through simple cellular interactions the immune system could be activated with high precision.

The CFF proposed that cellular frustrated interactions could be crucial to build a system of cells in interaction whose dynamics would be extremely sensible to changes in population configurations [43]. This happened because generalized kinetic proofreading mechanisms were identified [43, 44, 61] making each T cell a context discriminator (or detector). Indeed, in [61] it was shown that in CFSs the dissociation constant that distinguishes alternative pathways before major signals are produced [62] is renormalized by factors related to cell frequencies in the population. The production of major signals was straightforwardly related to long-lived conjugations and it was proposed that the focus for understanding how the adaptive immune system works should be placed on which cells establish those long-lived conjugations.

In [50] it was shown that, despite the fact that CFSs are very sensitive to changes, a set of configurations could be defined to be tolerated. To select which configurations to tolerate, a new organization principle was proposed involving all interaction lists (ILists) in the population. The principle of maximal frustration suggested that ILists were selected to reduce conjugation lifetimes for all cells in

the population and in this way taking into account the information in normal (self) configurations.

The previous results motivated a research program that had as main goal identifying the simplest frustrated system that could explain fundamental observations and simultaneously why the immune system is competent at protecting the host. In [49] it was shown that the simplest system could involve only APCs and T cells. This would be enough to explain why the immune system accurately detects nonself ligands in the periphery. However, to accomplish this, negative and positive selection would be required. Furthermore, costimulation and anergy would be necessary to improve the accuracy. These results are consistently accounted within the CFF scenario, as summarized in Figure 5.2.

By raising the question "Can the immune system perform a t-test?" this work calls attention to the fact that performing self-nonself discrimination could be largely insufficient to detect anomalies in the immune system. In statistics it is well known that the null hypothesis can be rejected even in the absence of outliers. It can be the presence of multiple small biases that signal an anomaly. Here we showed that CFSs could perform the same tasks as todays most popular statistical and machine learning techniques, and with comparable, if not greater, accuracies. This can have profound immunological consequences since it was shown how the immune system can monitor simultaneously the presence of nonself ligands and the presence of combinations of frequently presented ligands.

The current results are part of a framework, a different mind setting to model cellular interactions and detect changes in the displayed information. These results suggest new directions of research, many of them naturally linked to immunology. Indeed, B cell activation was not considered, although the activation of B cells by T helper cells in lymph nodes [63, 64] may likely be explained using a similar modelling approach. Investigating how B cells are activated can also be extremely relevant because the considerable data that exists today [65, 66] makes it possible to confront theory and observation [67–69]. In this respect, it would be important to include the impact of effector functions like clonal expansion and the introduction of regulatory T cells to study how the immune response regulates homeostasis. To understand the immune system The CFF proposes finding the simplest model that is capable of performing the most accurate anomaly detection under plausible immunological conditions. For instance, in [50] we showed that three cell types (T cells, APCs and Tregs) could build a frustrated dynamics with detection capabilities. However, in [49] we showed that to achieve perfect self-nonself discrimination, Tregs were not required. Likewise, here we show that a model with only T cells and APCs is enough to explain how abnormal detection capabilities can be achieved. Therefore, so far, the CFF suggests that Tregs play no essential role for achieving good detection

capabilities. This conclusion applies at least under static conditions. However, this conclusion may change if other requirements are imposed on how the information displayed evolves in time. This will be a matter of discussion in a future publication.

Another important conclusion resulting from this work concerns the role played by frequent and rare ligands for activation of the immune system. In [49] it was shown that nonself ligands - the rarest ligands possible - trigger specific cell activations, when these ligands appear in the system. By contrast, in abnormal-self discrimination, long-lived conjugations do not necessarily involve rare ligands and arise due to the absence of sets of frequent ligands. Therefore, the immune system monitors the two types of ligands in the system and in different ways.

Another point worth noting is that the results reported in this paper show how the immune system can be triggered to respond to changes in the displayed information. However, nothing is said concerning how responses can be set to resolve the causes of challenges. This is another topic to be discussed elsewhere and it will require an enlarged systemic approach to model the immune system.

How the real information displayed by APCs shapes the information contained in T cell ILists (i.e., T cell receptors) is a topic that should be further investigated taking into account the real aminoacid frequencies [54], details on recombinations events [70] or details on the interaction energies between APCs and T cell receptors [71].

Another issue that could be interesting exploring is the relation between CFF to other frustration models used in immunology [72–75] and possible connections to neuronal networks [76]. In particular, it would be interesting how the two classes of models compare at performing equivalent immune protection tasks.

Finally, the ideas explored in this article can be further developed in other directions. It would be interesting to consider whether the concept-drift idea [77] proposed to address anomaly detection in non-stationary datasets in artificially immune systems could also be useful, to explain the adaptive immune system. Another possible direction, would be to understand how danger theory ideas [25, 78] could be compatible with the CFFs. This work also suggests several ideas for research in machine learning, such as extending the current results to unsupervised learning [79] and even deep learning strategies in anomaly detection [80].

## 5.7 Methods

### Agents Based Model

The agent based model used in this article can be summarized as follows. The model considers two cell types, APCs and T cells, each having two subtypes, $I$ and $II$, with $N/2$ cells each. APCs favour interactions with T cells of the same subtype.

Each APC presents a single ligand, $p_i$. For an APC with index $i$, $p_i$ is a real number between $i$ and $i+1$. Each T cell has a connectivity list with $K$ different APCs with whom it can interact with. T cells map the information sensed on each APCs into only two possible signals (or ligands), $F_i$ or $R_i$, standing respectively for frequent or rare. Different specific mapping rules can be designed. In this article each T cell either senses rare ligands on the right or the left tail. For each T cell with index $j$ a discrimination parameter $v_j$ is drawn from a uniform distribution between 0 and $v_{max}$. Designate $f_i^0(x)$ the frequency of occurrence of ligand $p_x$ in self-configurations presented during education. Define left and right discrimination ligand thresholds, $p_i^L$ and $p_i^R$, such that:

$$\int_i^{p_i^L} f_i^0(x)dx = v_j, \qquad \int_{p_i^R}^{i+1} f_i^0(x)dx = v_j \tag{5.8}$$

Then, a ligand $p_i$ is mapped onto a rare ligand $R_i$ if $p_i < p_i^L$ for T cells sensing rare ligands on the left tail, or $p_i > p_i^R$ if they sense them on the right tail. In the DinBs case study, all T cells used the same mapping and therefore each APC can be associated to a rare of frequent ligand. In this case, $p_i$'s were selected so that the number of rare ligands in each configuration was fixed. In more general cases, $p_i$ derive from the dataset.

Each T cell arranges $F_i$ and $R_i$ ligands in ordered interaction lists (ILists), prioritizing interactions with top ranked ligands (or signals). Modelling of cellular interactions assumes a discrete time dynamics where, at each time step, a randomly drawn cell is put in interaction with a cell from the other cell type and belonging to its connectivity list. A new conjugation is established whenever the two cells that are put in interaction, prioritize this interaction. In that case, if they were already conjugated, former conjugations are terminated and the duration of that conjugation is registered (see SM1).

Negative selection in thymic repertoire education is modelled eliminating T cells that remain conjugate for a time longer than a threshold lifetime $\tau_n$ (see SM1). A new cell is introduced in the population with the same connectivity list but with a randomly drawn IList. If after $W_\tau$ iterations (typically 10000 iterations) no cells exceeded the threshold time, then $\tau_n$ is updated to the largest conjugation time in the last $W_\tau$ iterations and the T cell population is registered. Every $T_s$ iterations (typically 100 iterations), ligands presented by APCs present information from a different sample (configuration). When $\tau_n$ stops changing for at least $10^6$ iterations, education stops, the last recorded population is added to the T cell repertoire and education of a new T cell population is started, maintaining each T cell connectivity list but randomly drawing its IList.

After creating a repertoire of educated T cells, the population enters the calibra-

tion stage (see SM1). This should correspond to the final stage of T cell repertoire education. At this stage cells engage in the same decision dynamics as before, except that anergy is introduced, so that T cells conjugated for a time longer than $\tau_A$ terminate their conjugations and are replaced by other cells with the same connectivity in the repertoire. In our results we used $\tau_A = 5$. During calibration only self-information is presented. This is modelled by gathering information for presentation by APCs from the same set of samples that were used in the education stage. During calibration the decision dynamics is run for $W_c$ iterations for each sample (typically $W_c = 10^4$ iterations) and the number of conjugations lasting longer than an activation lifetime $\tau_{act}$, involving APC with index $i$ when sample $s$ is presented, $c^0_{i,s}(\tau_{act})$, are registered. Defining the ordered vector $c^0_{i,(j)}(\tau_{act})$, such that $c^0_{i,(j)}(\tau_{act}) \geq c^0_{i,(j+1)}(\tau_{act}) \forall j$, then an activation threshold is established by $n^0_i(\tau_{act}) = c^0_{i,(x)}(\tau_{act})$ where $x = N^c_s \times f$, with $N^c_s$ the number of samples used during the calibration and $f$ is a real number between 0 and 1. Typically we used $f = 0.1$, and hence the 10% biggest number of conjugations lasting a time larger than $\tau_{act}$ in a sample was considered. The activation reference time is chosen to be equal to the largest conjugation time in the calibration, i.e., $\tau_{act} = \tau_A$.

To model cellular activation in lymph nodes the decision dynamics is run in the same conditions as in the calibration stage. To analyse the performance of the model in detecting anomalies, APCs display either information from a self-dataset, or from a nonself or abnormal-self dataset. Several examples are illustrated in the Results section. The cellular response to the information displayed by sample $s$ is calculated according to:

$$R_s = \sum_i c_{i,s}(\tau_{act}) - n^0_i(\tau_{act}))/c_{i,s}(0) \times \theta(c_{i,s}(\tau_{act}) - n^0_i(\tau_{act}) \tag{5.9}$$

where $\theta$ is the Heaviside function. Thus the cellular response sums the increments on the number of long conjugations relatively to the calibration stage. In this expression a subtle contribution from positive selection was taken. Instead of simulating the process explicitly, as in [49], we normalized the number of conjugations in the time interval $W_c$ therefore accounting for the impact of positive selection as depicted in Figure 5.2.

To quantify the detection accuracy we compute the true positive rate for a fixed false positive rate, $FPR$. To achieve this we create and ordered vector of population responses to the $N^{d,s}_s$ samples in the detection stage displaying self-information, $R^s_{(i)}$, such that $R^s_{(i)} \geq R^s_{(i+1)} \forall i$ and find $R^s_x$, where $x = N^{d,s}_s \times FPR$. Then the true positive rate becomes $TPR = \#\{R^{ns}_s : R^{ns}_s > R^s_x\}/N^{d,ns}_s$, where $R^{ns}_s$ are the population responses to samples displaying nonself or abnormal self information and $N^{d,ns}_s$ is the number of nonself or abnormal-self samples presented during the

detection stage. The true positive rate is thus equal to the fraction of samples displaying nonself or abnormal self information leading to cellular responses greater than $R_x^s$.

## Applications - dataset generation

Three synthetic datasets were generated to evaluate the performance of cellular frustrated systems (CFSs). All datasets are comprised of samples with 80 elements ($N = 80$).

The first two datasets were used for comparison with conventional statistical tests (t-student and KS-test). In the first case, self-samples were drawn from normal distributions with $\mu_S = 50$ and $\sigma = 10$. Abnormal self-configurations were generated from normal distributions with the same standard deviation but with $\mu_{NS} = 50 \pm \Delta$, where $\Delta = 1, 2$ or 4 depending on the example. In the second case, samples were drawn from lognormal instead of normal distributions. Means and standard deviations were changed so that lognormal distributions maintained the same means and standard deviation.

The third data set was obtained as in the first dataset, however no ordering was applied. Furthermore, each element in abnormal-self samples could be drawn from a normal distributions deviated to either side. The example considered in this case used $\Delta = 4$.

In each comparison, sets of samples were comprised of $N_s^c = 1000$ self samples for training, $N_s^{d,s} = 1000$ self samples for evaluation of the false positive rate, and $N_s^{d,ns} = 1000$ to evaluate the true positive rate. When comparisons with data mining classification algorithms were considered, a supplementary set with 1000 abnormal-self samples with $\mu_{NS} = 50 + \Delta^0$ were used for training. Two cases were studied: $\Delta^0 = 2$ or $\Delta^0 = 8$ (Figure 5.16).

## Applications - Statistical and data mining methods

Conventional two sided t and KS tests used the R implementation with known $\sigma$. Packages available in R, randomForests [81] and e1071 (libSVM) [82, 83] were used to apply two data-mining algorithms, respectively random forests and support vector machines. Two versions of support vector machines were considered, one-class SVM with a polynomial Kernel, and two-class SVMs with a RBF (radial basis function) Kernel. Both methods were applied using default packages parameters. This is particularly acceptable in case of anomaly detection, since knowledge of the abnormal class is not available. To compute average performances all tests were repeated ten times.

## 5.8 Supplementary Materials

### SM1-Numerical algorithm flowchart

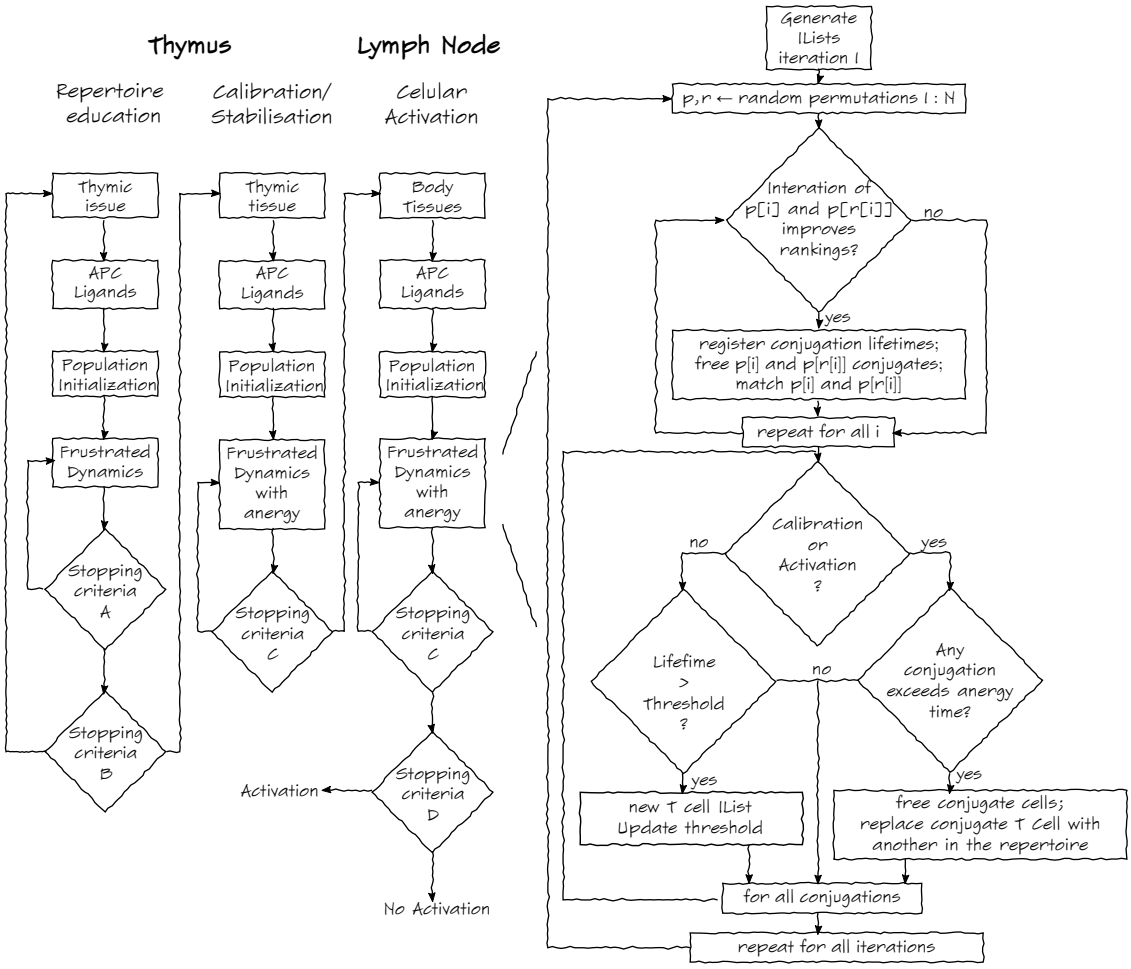


Figure 5.17: Flowchart with the main steps in the numerical simulations used in this work. In repertoire education negative selection is used to order T cell lLists until a stopping criteria A is met. This process is then repeated for several T cell populations (stopping criteria B). After repertoire education the system enters the calibration/stabilisation stage. The frustrated dynamics with anergy is run for $W_c$ iterations (stopping criteria C), typically $W_c = 10^4$, to establish the characteristic frequency of conjugations lasting for a time $\tau$. The same dynamics is run in the lymph node for cellular activation. However, now any samples can be presented (self or abnormal self). During cellular activation if a T cell exceeds the characteristic frequency of conjugations lasting for a time $\tau$ an immune response is mounted (stopping criteria D).

## SM2-Partially ordered interaction lists organization

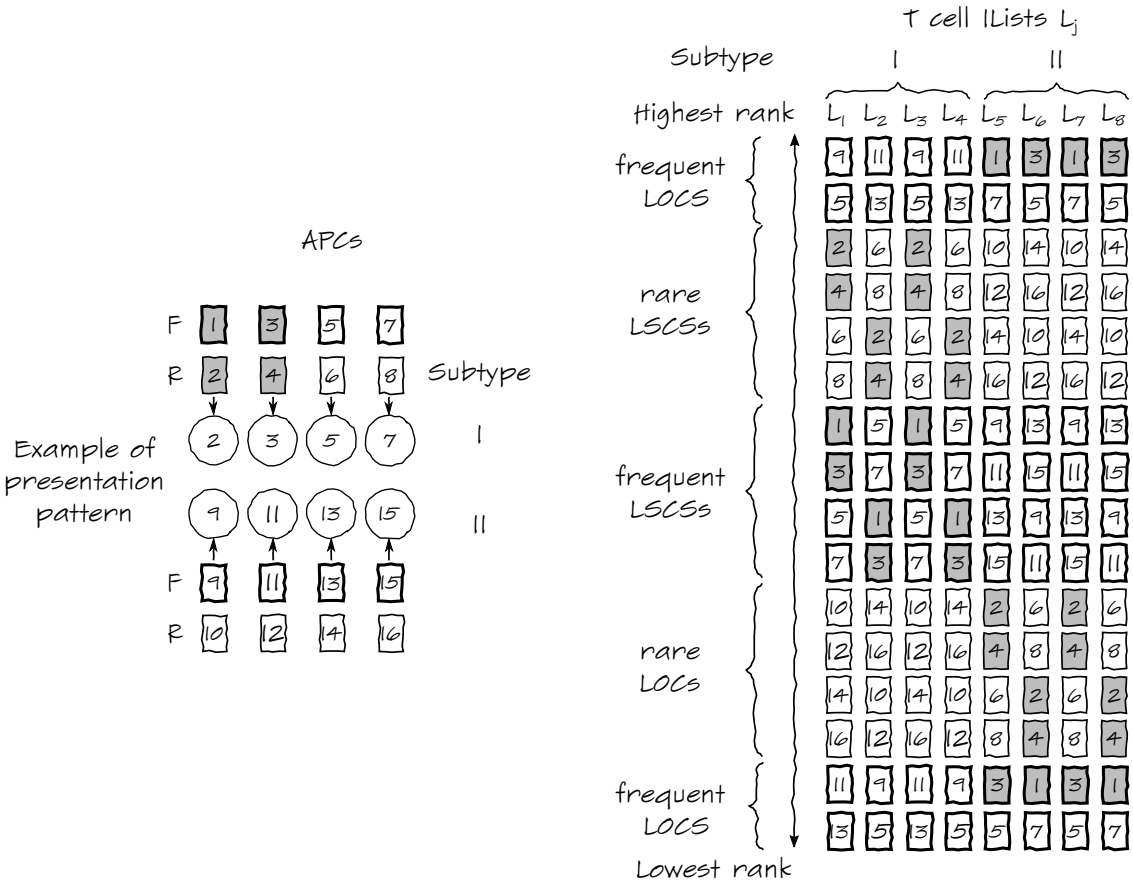


Figure 5.18: Illustration of the generation scheme used to build T cell partially ordered ILists. *Left*) Four APCs of two subtypes, I and II, can present either frequent or rare ligands (odd and even numbers, enclosed in darker or lighter boxes). In the configuration shown, the first APC presents a rare ligand (2) while the remaining APCs present frequent ligands. In this simple population with only four APCs of each subtype, the first two subtype I APCs belong to a same block (filled boxes). *Right*) Partially ordered T cell ILists with 2 frequent LOCS on top positions. Ligands are ranked sequentially. On top positions appear frequent LOCSs; then rare LSCSs; then frequent LSCSs; then rare LOCS; finally the remaining frequent LOCSs. Frequent LOCS use a different ordering scheme than the remaining ligands. Instead of being consecutively listed, frequent LOCSs are listed in pairs: ligand 7 is always listed after ligand 1; ligand 5 is always listed after ligand 3; ligand 15 is always listed after ligand 9; ligand 13 is always listed after ligand 11.

## SM3-Companion figure for the discussion of the decrease in the discrimination when a population is educated with $N_r^0$ rare ligands and $N_r^0 + 1$ rare ligands are presented.

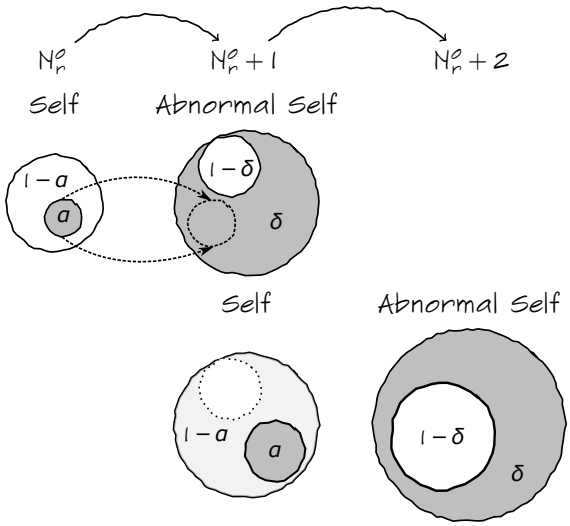


Figure 5.19: Evolution of the number of activated configurations when the number of rare ligands is sequentially incremented. On the top left, the set of self configurations with $N_r^0$ rare ligands are displayed. From these, a fraction $\alpha$ comprises those configurations leading to activation of T cells with the total highest magnitude. When an additional rare ligand the later configurations will remain activated and a fraction of the remaining $(1 - \alpha)$ configurations will also lead to configurations with strong activations (indicated by dashed arrows and dashed circle). Overall, the fraction of the activated configurations will be $\delta$. If one now considers that self configurations have $N_r^0 + 1$ rare ligands, since $\alpha < \delta$ only a subset of the later configurations – most activated of them – will lead to activated configurations. In this case, the magnitude required for activation is higher. Using the previous reasoning, that only a fraction of the least activated configurations become activated when an additional is displayed, then, when $N_r^0 + 2$ ligands are displayed, the most activated configurations with $N_r^0 + 1$ ligands will lead to activated configurations with $N_r^0 + 2$ ligands; A fraction of configurations with intermediate activation (represented in light grey) will also be activated; Only a very minor fraction of configurations in white will become activated when $N_r^0 + 2$ ligands are presented. Therefore, discrimination becomes poorer when the number of rare ligands displayed during education is higher and a rare is added for discrimination.

## SM4-Graph showing the evolution of the number of frequent LOCS in top positions in T cell ILists in the DinBs case study

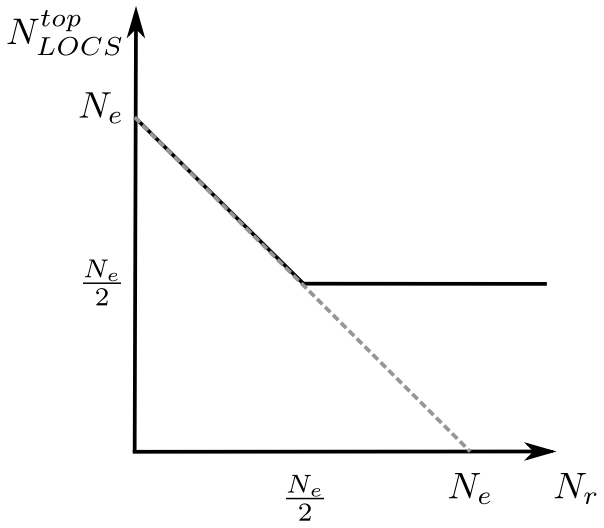


Figure 5.20: Minimum number of frequent LOCS in a T cell IList for self (solid) and abnormal-self (dashed) configurations Increasing the number of rare ligands has a different impact on the number of frequent LOCS in top positions in T cell ILists for self or abnormal-self configurations in the DinBs case study. The organization in T cell ILists guarantees that half of the top ligands belong to a block not displaying rare ligands in self configurations. However, for nonself configurations some ILists may have all frequent LOCSs on top positions absent.

## 5.9 Bibliography

[1] Bruno Filipe Faria, Patrícia Mostardinha, and Fernão Vístulo de Abreu. Can the Immune System Perform a t-Test? *PLOS ONE*, 12(1), jan 2017. doi: 10.1371/journal.pone.0169464.

[2] N. K. Jerne. Towards a network theory of the immune system. *Annales d'immunologie*, 125C(1-2):373–389, January 1974. ISSN 0300-4910.

[3] I.R. Cohen. *Tending Adam's Garden: Evolving the Cognitive Immune Self*. Academic Press, Waltham, MA, 2000. ISBN 9780121783556.

[4] F.M. Burnet and F. Fenner. *The Production of Antibodies*. Macmillan, Melbourne, 1949.

[5] M. Cohn and R. E. Langman. To be or Not to be Ridded? - That is the Question Addressed by the Associative Antigen Recognition Model. *Scandinavian Journal of Immunology*, 55(4):318–323, apr 2002. ISSN 1365-3083.

[6] Rob J. De Boer and Alan S. Perelson. How diverse should the immune system be? *Proceedings of the Royal Society of London B: Biological Sciences*, 252 (1335):171–175, 1993.

[7] Stephanie Forrest, Alan S. Perelson, Lawrence Allen, and Rajesh Cherukuri. Self-Nonself Discrimination in a Computer. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, SP '94, pages 202–212, Washington, DC, USA, 1994. IEEE Computer Society.

[8] Franco Celada and Philip E. Seiden. A computer model of cellular interactions in the immune system. *Immunology Today*, 13(2):56–62, 1992. ISSN 0167-5699.

[9] T. Stibor, J. Timmis, and C. Eckert. On the Use of Hyperspheres in Artificial Immune Systems as Antibody Recognition Regions. In *Proceedings of the 5th International Conference on Artificial Immune Systems (ICARIS-2006)*, volume 4163 of *Lecture Notes in Computer Science*, pages 215–228, Oeiras, Portugal, 2006. Springer-Verlag.

[10] Zhou Ji and Dipankar Dasgupta. Revisiting Negative Selection Algorithms. *Evolutionary Computation*, 15(2):223–251, June 2007. ISSN 1063-6560.

[11] Alfred I. Tauber. Immunology and the enigma of selfhood. In M. Norton Wise, editor, *Growing Explanations: Historical Perspectives on Recent Science*, chapter 7, pages 199–221. Durham: Duke University Press, Durham NC, 2004.

[12] A. Coutinho, L. Forni, D. Holmberg, F. Ivars, and N. Vaz. From an Antigen-Centered, Clonal Perspective of Immune Responses to an Organism-Centered, Network Perspective of Autonomous Activity in a Self-Referential Immune System. *Immunol Rev*, 79(1):151–168, jun 1984.

[13] Antonio Coutinho, Michel D Kazatchkine, and Stratis Avrameas. Natural autoantibodies. *Current Opinion in Immunology*, 7(6):812–818, 1995. ISSN 0952-7915.

[14] Klaus Eichmann et al. *The network collective: Rise and fall of a scientific paradigm*. Birkhauser Verlag AG, 2008. ISBN 9783764383732.

[15] Thomas Kieber-Emmons, Bejatohlah Monzavi-Karbassi, Anastas Pashov, Somdutta Saha, Ramachandran Murali, and Heinz Kohler. The Promise of the Anti-Idiotype Concept. *Frontiers in Oncology*, 2(196), 2012. ISSN 2234-943X.

[16] Francisco J. Varela and Antonio Coutinho. Second generation immune networks. *Immunology Today*, 12(5):159–166, 1991. ISSN 0167-5699.

[17] Robert Schulz, Benjamin Werner, and Ulrich Behn. Self tolerance in a minimal model of the idiotypic network. *Frontiers in Immunology*, 5(86), 2014. ISSN 1664-3224.

[18] Elena Agliari, Adriano Barra, Gino Del Ferraro, Francesco Guerra, and Daniele Tantari. Anergy in self-directed B lymphocytes: A statistical mechanics perspective. *Journal of Theoretical Biology*, 375:21–31, 2015. ISSN 0022-5193. Theories and Modeling of Autoimmunity.

[19] R.E. Langman and M. Cohn. A minimal model for the self-nonself discrimination: a return to the basics. *Seminars in Immunology*, 12(3):189–195, 2000. ISSN 1044-5323.

[20] Alfred I. Tauber. The immune self: theory or metaphor? *Immunology Today*, 15(3):134–136, 1994. ISSN 0167-5699.

[21] Sol Efroni and Irun R. Cohen. Simplicity belies a complex system: a response to the minimal model of immunity of Langman and Cohn. *Cellular Immunology*, 216(1–2):23–30, 2002. ISSN 0008-8749.

[22] Z. Dembic. On Recognizing "Shades-of-Gray" (Self-Nonself Discrimination) or "Colour" (Integrity Model) by The Immune System. *Scand J Immunol*, 78(4): 325–338, sep 2013.

[23] Sir Frank Macfarlane Burnet et al. *The clonal selection theory of acquired immunity*, volume 3. Vanderbilt University Press Nashville, 1959.

[24] Johnny Kelsey and Jon Timmis. *Immune Inspired Somatic Contiguous Hypermutation for Function Optimisation*, pages 207–218. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-45105-1.

[25] Feng Gu, Julie Greensmith, and Uwe Aickelin. Theoretical formulation and analysis of the deterministic dendritic cell algorithm. *Biosystems*, 111(2): 127–135, 2013. ISSN 0303-2647.

[26] Ilhan Aydin, Mehmet Karakose, and Erhan Akin. Chaotic-based hybrid negative selection algorithm and its applications in fault and anomaly detection. *Expert Systems with Applications*, 37(7):5285–5294, 2010. ISSN 0957-4174.

[27] D. Dasgupta, K. KrishnaKumar, D. Wong, and M. Berry. Negative Selection Algorithm for Aircraft Fault Detection. In *Lecture Notes in Computer Science*, pages 1–13. Springer Science + Business Media, 2004.

[28] D. Dasgupta and F. Gonzalez. An immunity-based technique to characterize intrusions in computer networks. *IEEE Transactions on Evolutionary Computation*, 6(3):281–291, jun 2002.

[29] Vladimir Golovko, Sergei Bezobrazov, Pavel Kachurka, and Leanid Vaitsekhovich. *Neural Network and Artificial Immune Systems for Malware and Network Intrusion Detection*, pages 485–513. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-05179-1.

[30] Ismaila Idris and Ali Selamat. Improved email spam detection model with negative selection algorithm and particle swarm optimization. *Applied Soft Computing*, 22:11–27, 2014. ISSN 1568-4946.

[31] Mario Pavone, Giuseppe Narzisi, and Giuseppe Nicosia. Clonal selection: an immunological algorithm for global optimization over continuous spaces. *Journal of Global Optimization*, 53(4):769–808, jun 2011.

[32] F. Campelo, F.G. Guimaraes, H. Igarashi, and J.A. Ramirez. A clonal selection algorithm for optimization in electromagnetics. *IEEE Transactions on Magnetics*, 41(5):1736–1739, may 2005.

[33] B. Babayigit, A. Akdagli, and K. Guney. A Clonal Selection Algorithm for null Synthesizing of Linear Antenna Arrays by Amplitude Control. *Journal of Electromagnetic Waves and Applications*, 20(8):1007–1020, 2006.

[34] A.M. Whitbrook, U. Aickelin, and J.M. Garibaldi. Idiotypic Immune Networks in Mobile-Robot Control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(6):1581–1598, dec 2007.

[35] Yen-Nien Wang, Hao-Hsuan Hsu, and Chun-Cheng Lin. *Artificial Immune Algorithm Based Obstacle Avoiding Path Planning of Mobile Robots*, pages 859–862. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-31858-3.

[36] Thomas Stibor, Philipp Mohr, Jonathan Timmis, and Claudia Eckert. Is Negative Selection Appropriate for Anomaly Detection? In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, GECCO '05, pages 321–328, New York, NY, USA, 2005. ACM. ISBN 1-59593-010-8.

[37] Amanda V. Gett, Federica Sallusto, Antonio Lanzavecchia, and Jens Geginat. T cell fitness determined by signal strength. *Nature Immunology*, 4(4):355–360, mar 2003. doi: 10.1038/ni908. URL `http://dx.doi.org/10.1038/ni908`.

[38] Philippe Bousso. T-cell activation by dendritic cells in the lymph node: lessons from the movies. *Nature Reviews Immunology*, 8(9):675–684, sep 2008.

[39] A.K. Abbas and A.H. Lichtman. *Basic Immunology: Functions and Disorders of the Immune System*. Elsevier/Saunders, Philadelphia, PA, 2010. ISBN 9781416055693.

[40] R. N. Germain. The Art of the Probable: System Control in the Adaptive Immune System. *Science*, 293(5528):240–245, jul 2001. doi: 10.1126/science.1062946. URL `http://dx.doi.org/10.1126/science.1062946`.

[41] David Depoil, Rossana Zaru, Martine Guiraud, Anne Chauveau, Julie Harriague, Georges Bismuth, Clemens Utzny, Sabina Müller, and Salvatore Valitutti. Immunological Synapses Are Versatile Structures Enabling Selective T Cell Polarization. *Immunity*, 22(2):185–194, 2005. ISSN 1074-7613.

[42] Joost B. Beltman, Athanasius F.M. Marée, Jennifer N. Lynch, Mark J. Miller, and Rob J. de Boer. Lymph node topology dictates T cell migration behavior. *The Journal of Experimental Medicine*, 204(4):771–780, mar 2007.

[43] F. Vístulo de Abreu, E. N. M. Nolte-'Hoen, C. R. Almeida, and D. M. Davis. Cellular Frustration: A New Conceptual Framework for Understanding Cell-mediated Immune Responses. In *Proceedings of the 5th International Conference on Artificial Immune Systems*, ICARIS'06, pages 37–51, Berlin, Heidelberg, 2006. Springer-Verlag.

[44] C.R. Almeida and F.V. de Abreu. Dynamical instabilities lead to sympatric speciation. *Evolutionary Ecology Research*, 5(5):739–757, 2003.

[45] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[46] Robert W. Irving. The cycle roommates problem: a hard case of kidney exchange. *Information Processing Letters*, 103(1):1–4, 2007. ISSN 0020-0190.

[47] Yunan Gu, Yanru Zhang, Miao Pan, and Zhu Han. Matching and Cheating in Device to Device Communications Underlying Cellular Networks. *IEEE Journal on Selected Areas in Communications*, 33(10):2156–2166, oct 2015.

[48] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Cambridge, MA, USA, 1989. ISBN 0-262-07118-5.

[49] P. Mostardinha and F. Vístulo de Abreu. Positive and negative selection, self-nonself discrimination and the roles of costimulation and anergy. *Scientific Reports*, 2:769, oct 2012. ISSN 2045-2322.

[50] F. Vístulo de Abreu and P. Mostardinha. Maximal frustration as an immunological principle. *Journal of The Royal Society Interface*, 6(32):321–334, 2009. ISSN 1742-5689.

[51] G. Parisi. Two signals from B cells control the expansion of T cells: only one is immunologically specific. *Annales de l'Institut Pasteur / Immunologie*, 139(2):177–185, mar 1988. doi: 10.1016/0769-2625(88)90039-6. URL `http://dx.doi.org/10.1016/0769-2625(88)90039-6`.

[52] Julie M. Curtsinger and Matthew F. Mescher. Inflammatory cytokines as a third signal for T cell activation. *Current Opinion in Immunology*, 22(3):333–340, jun 2010.

[53] Grégoire Altan-Bonnet and Ronald N Germain. Modeling T Cell Antigen Discrimination Based on Feedback Control of Digital ERK Responses. *PLoS Biology*, 3(11):e356, oct 2005. doi: 10.1371/journal.pbio.0030356. URL `http://dx.doi.org/10.1371/journal.pbio.0030356`.

[54] Andrej Košmrlj, Arup K. Chakraborty, Mehran Kardar, and Eugene I. Shakhnovich. Thymic Selection of T-Cell Receptors as an Extreme Value Problem. *Physical Review Letters*, 103(6), aug 2009. doi: 10.1103/PhysRevLett.103.068103. URL `http://dx.doi.org/10.1103/PhysRevLett.103.068103`.

[55] Arup K. Chakraborty and Andrej Košmrlj. Statistical Mechanical Concepts in Immunology. *Annual Review of Physical Chemistry*, 61(1):283–303, mar 2010. doi: 10.1146/annurev.physchem.59.032607.093537. URL `http://dx.doi.org/10.1146/annurev.physchem.59.032607.093537`.

[56] Dennis L. Chao, Miles P. Davenport, Stephanie Forrest, and Alan S. Perelson. A stochastic model of cytotoxic T cell responses. *Journal of Theoretical Biology*, 228(2):227–240, may 2004. doi: 10.1016/j.jtbi.2003.12.011. URL `http://dx.doi.org/10.1016/j.jtbi.2003.12.011`.

[57] J. Neyman and E. Pearson. *Sufficient statistics and uniformly most powerful tests of statistical hypotheses*. Statist, 1936.

[58] J. Neyman and E. Pearson. *Contributions to the Theory of Testing Statistical Hypotheses*. Statist, 1938.

[59] Maxwell L. King. The power of Students's t test: can a non-similar test do better? *Australian Journal of Statistics*, 32(1):21–27, mar 1990.

[60] Rand R. Wilcox. *Fundamentals of Modern Statistical Methods*. Springer Science + Business Media, 2010.

[61] André M. Lindo, Bruno F. Faria, and Fernão V. de Abreu. Tunable kinetic proofreading in a model with molecular frustration. *Theory in Biosciences*, 131 (2):77–84, 2012. ISSN 1611-7530.

[62] T W McKeithan. Kinetic proofreading in T-cell receptor signal transduction. *Proceedings of the National Academy of Sciences*, 92(11):5042–5046, 1995.

[63] Christopher C. Goodnow, Jonathon Sprent, Barbara Fazekas de St Groth, and Carola G. Vinuesa. Cellular and genetic mechanisms of self tolerance and autoimmunity. *Nature*, 435(7042):590–597, jun 2005. doi: 10.1038/nature03724. URL `http://dx.doi.org/10.1038/nature03724`.

[64] Christopher C. Goodnow, Carola G. Vinuesa, Katrina L. Randall, Fabienne Mackay, and Robert Brink. Control systems and decision making for antibody production. *Nature Immunology*, 11(8):681–688, jul 2010. doi: 10.1038/ni.1900. URL `http://dx.doi.org/10.1038/ni.1900`.

[65] J. A. Weinstein, N. Jiang, R. A. White, D. S. Fisher, and S. R. Quake. High-Throughput Sequencing of the Zebrafish Antibody Repertoire. *Science*, 324 (5928):807–810, may 2009. doi: 10.1126/science.1170020. URL `http://dx.doi.org/10.1126/science.1170020`.

[66] Irun R. Cohen. Autoantibody repertoires, natural biomarkers, and system controllers. *Trends in Immunology*, 34(12):620–625, dec 2013. doi: 10.1016/j.it.2013.05.003. URL `http://dx.doi.org/10.1016/j.it.2013.05.003`.

[67] Asaf Madi, Sharron Bransburg-Zabary, Dror Y. Kenett, Eshel Ben-Jacob, and Irun R. Cohen. The Natural Autoantibody Repertoire in Newborns and Adults. pages 198–212, 2012. doi: 10.1007/978-1-4614-3461-0_15.

[68] Sharron Bransburg-Zabary, Dror Y. Kenett, Gittit Dar, Asaf Madi, Yifat Merbl, Francisco J. Quintana, Alfred I. Tauber, Irun R. Cohen, and Eshel Ben-Jacob. Individual and meta-immune networks. *Physical Biology*, 10 (2):025003, mar 2013. doi: 10.1088/1478-3975/10/2/025003. URL `http://dx.doi.org/10.1088/1478-3975/10/2/025003`.

[69] T. Mora, A. M. Walczak, W. Bialek, and C. G. Callan. Maximum entropy models for antibody diversity. *Proceedings of the National Academy of Sciences*, 107(12):5405–5410, mar 2010. doi: 10.1073/pnas.1001705107. URL `http://dx.doi.org/10.1073/pnas.1001705107`.

[70] A. Murugan, T. Mora, A. M. Walczak, and C. G. Callan. Statistical inference of the generation probability of T-cell receptors from sequence repertoires. *Proceedings of the National Academy of Sciences*, 109(40):16161–16166, sep 2012. doi: 10.1073/pnas.1212755109. URL `http://dx.doi.org/10.1073/pnas.1212755109`.

[71] Nigel J. Burroughs, Rob J. de Boer, and Can Keşmir. Discriminating self from nonself with short peptides from large proteomes. *Immunogenetics*, 56 (5):311–320, jul 2004. doi: 10.1007/s00251-004-0691-0. URL `http://dx.doi.org/10.1007/s00251-004-0691-0`.

[72] G. Parisi. A simple model for the immune network. *Proceedings of the National Academy of Sciences*, 87(1):429–433, jan 1990. doi: 10.1073/pnas.87.1.429. URL `http://dx.doi.org/10.1073/pnas.87.1.429`.

[73] Elena Agliari, Adriano Barra, Francesco Guerra, and Francesco Moauro. A thermodynamic perspective of immune capabilities. *Journal of Theoretical Biology*, 287:48–63, oct 2011. doi: 10.1016/j.jtbi.2011.07.027. URL `http://dx.doi.org/10.1016/j.jtbi.2011.07.027`.

[74] E. Agliari, A. Annibale, A. Barra, A. C. C. Coolen, and D. Tantari. Immune networks: multitasking capabilities near saturation. *Journal of Physics A: Mathematical and Theoretical*, 46(41):415003, sep 2013. doi: 10.1088/1751-8113/46/41/415003. URL `http://dx.doi.org/10.1088/1751-8113/46/41/415003`.

[75] E. Agliari, A. Annibale, A. Barra, A. C. C. Coolen, and D. Tantari. Immune networks: multi-tasking capabilities at medium load. *Journal of Physics A:*

*Mathematical and Theoretical*, 46(33):335101, jul 2013. doi: 10.1088/1751-8113/46/33/335101. URL http://dx.doi.org/10.1088/1751-8113/46/33/335101.

[76] Elena Agliari, Adriano Barra, Andrea Galluzzi, Francesco Guerra, and Francesco Moauro. Multitasking Associative Networks. *Physical Review Letters*, 109(26), dec 2012. doi: 10.1103/PhysRevLett.109.268101. URL http://dx.doi.org/10.1103/PhysRevLett.109.268101.

[77] Alaa Abi-Haidar and Luis M. Rocha. *Adaptive Spam Detection Inspired by a Cross-Regulation Model of Immune Dynamics: A Study of Concept Drift*, pages 36–47. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-85072-4.

[78] Polly Matzinger. The Danger Model: A Renewed Sense of Self. *Science*, 296 (5566):301–305, 2002. ISSN 0036-8075.

[79] Markus Goldstein and Seiichi Uchida. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLoS ONE*, 11(4): 1–31, 04 2016.

[80] Min-Joo Kang and Je-Won Kang. Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security. *PLoS ONE*, 11(6):1–17, 06 2016.

[81] Andy Liaw and Matthew Wiener. Classification and Regression by randomForest. *R News*, 2(3):18–22, 2002.

[82] David Meyer and Technische Universität Wien. Support Vector Machines: The Interface to libsvm in package e1071. *R News*, pages 1–3, 2015.

[83] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3): 27:1–27:27, May 2011. ISSN 2157-6904.

CHAPTER 6

# Cellular frustration algorithms for anomaly detection applications[1]

Cellular frustrated models have been developed to describe how the adaptive immune system works. Given that the immune system's main function is to protect the own body, it has been hypothesized that these models could provide inspiration to develop new artificial intelligence algorithms for data mining applications. However, computational algorithms do not need to follow strictly the immunological reality. Here we investigate efficient implementation strategies of these immune inspired ideas for anomaly detection applications and compare the performance of cellular frustration algorithms with standard implementations of one-class support vector machines, using real data. Our results demonstrate that cellular frustration algorithms are robust and can be advantageous for semi-supervised anomaly detection applications.

## 6.1 Introduction

Cellular frustrated systems (CFSs) have been developed to model the adaptive immune system [2–4]. A crucial hypothesis raised in these works was that the immune system should be extremely competent at detecting deviations from its normal functioning, i.e., at performing anomaly detection. This hypothesis guided the search for the simplest model that, on one side, would be compatible with experimental observations in immunology and, on the other, could perform these immune function.

[1] chapter to be submitted as: B. F. Faria and F. Vístulo Abreu. Cellular frustration algorithms for anomaly detection applications. *(submitted)*, 2016

CFSs have the merits of taking assumptions that are reasonable from an immune system perspective. However, from a computational point of view this is not necessarily an advantage. Nature has certainly been capable of finding solutions for complex tasks through natural selection. However, these solutions need not be computationally efficient nor entirely focused in solving the task of interest to the computational scientist. Biological systems explored solutions that were accessible to the natural system and in agreement with the physical world constraints. However, biological systems have also to contend with a number of other challenges and hence had to find solutions that are also robust in face of these challenges. For instance, the immune system has to contend with cell number fluctuations, spatial constraints, or the available cellular interaction mechanisms.

In this paper we seek to develop efficient algorithms inspired in cellular frustrated systems. Instead of respecting the acceptable mechanisms from an immunological point of view, we relax constraints that can improve computational efficiency without compromising anomaly detection performance. To accomplish this, a new algorithm was developed with the important discrimination mechanisms in mind. As a result, the results reported here are important because they show that immune systems can be thought in more general terms.

This paper is organized as follows. In the following section we describe the different types of anomaly detection techniques and their relation to the cellular frustration framework (CFF). Then, we describe how anomaly detection is achieved within the CFF and define a cellular frustration algorithm (CFA) for anomaly detection applications in section 6.4. This algorithm gives special attention to the training stage, proposing a strategy that accelerates convergence. To gain a deeper understanding of the advantages of the new algorithm, a theoretical analysis is presented in section 6.5 showing that the new strategy converges faster than the immunological models proposed in [2–4]. This algorithm is tested with several datasets in section 6.6, and it is shown that the current training algorithm not only converges faster, but also, and more importantly, achieves good and more robust anomaly detection performances.

## 6.2 Brief Review of Anomaly Detection Approaches

Anomaly detection appears in the literature under several names, such as one-class learning, novelty detection, change detection, outlier detection or even failure detection. This shows the enormous relevance given to this topic by many different communities, each with different histories, techniques, terminologies and with

different applications in mind. Finding when a system changes its behaviour, or a dataset has elements that do not conform with the rest can be important to monitor motors and industrial processes [5, 6], to analyse human behaviour [7, 8], whole communities [9], to gain information from small datasets or address big data challenges [10], to protect single computers [11], computer networks [12], to make efficient learning algorithms [13] or to provide inspiration on how biological systems work [14, 15].

The anomaly detection topic has a considerable history, having been first addressed in statistics [16], and recently readdressed in the data mining field [17, 18]. Today, there are several data mining techniques addressing anomaly detection. They are generally divided in supervised, semi-supervised or unsupervised depending on the training required. Supervised techniques require training data with instances from the two categories, *normal* and *abnormal*. Semi-supervised techniques require only knowledge of normal instances. Unsupervised techniques use the available data to discern which instances are more likely to be distinct from the majority, i.e., anomalies.

Regardless of these distinctions, all these techniques try to detect a deviation from normality. How the different techniques establish the normality concept depends on the data and on the assumptions. The assumptions (e.g., the metrics in some distance based techniques) have a major impact in unsupervised techniques determining what can be detected. In other techniques this problem is not so dramatic, since training data can be used to tune parameters to minimize its impact. This makes unsupervised techniques less accurate, but simultaneously easier to use. Indeed, in many cases labelling data in categories is impossible. In this respect, semi-supervised techniques are a good alternative, since in many cases anomalies are rare and consequently their impact in training is small.

When labelled data is available, supervised techniques, also known as classification techniques, can be used. These techniques are usually the most accurate because the information of the sample's category is very effective. However, the accuracy in classification techniques is only warranted when both categories are balanced, i.e., both categories are equally represented in the dataset. For imbalanced datasets outcomes from anomaly and classification techniques can be distinctively different. This is especially true when the number of instances available from the anomalous class is not statistically significant. In basic terms this arises from the fact that classification techniques make the assumption that the available anomalous instances are typical examples of the anomalous class. This can be problematic for two reasons. Firstly because classification techniques can overweight anomalies which, in statistical terms, could be better seen as noise [17, 18]. Secondly because classification techniques may not recognize anomalies unavailable in training [17, 18].

A fundamental difference exists between anomaly detection techniques (either unsupervised or semi-supervised) and classification, which is that the former is concerned in establishing a predictive model of what is different relatively to what is *normal*, while classification techniques are concerned in defining the best model that is capable of distinguishing the two classes.

In practice, most techniques can be adapted to explore the different types of available data. For instance, support vector machines (SVMs) were initially developed by Vapnik for classification purposes [19, 20]. However, the scope of application of SVMs has been extended to tackle semi-supervised [21, 22] and unsupervised anomaly detection [23]. Still, SVMs were naturally defined as a classification technique and consequently, extensions required additional assumptions.

By contrast, cellular frustration algorithms (CFA) use training data to establish indicators of the normal class and therefore CFAs are naturally defined as semi-supervised techniques. However, we forsee extensions that may transform CFAs for supervised or unsupervised applications.

## 6.3 Brief Introduction to the Cellular Frustration Framework

The Cellular Frustration Framework is an agent based modelling approach that received inspiration from the stable marriage problem (SMP) introduced by Gale and Shapley [24]. In the SMP there are two types of agents, man and woman, and each agent has a preference list where an ordering of preferences for agents of the other type is listed. The aim is to marry men and women in a stable configuration, i.e., such that no man and woman in two distinct marriages prefer to be married with one another than with their current mates. This problem found applications in economy, since it could represent the labour market, with employers on one side and employees on the other. Both sides, gain by establishing stable matchings as they could waste their time otherwise.

For anomaly detection purposes the CFF proposes a different formulation of the problem. First the two agent types should have specific functions. One type of agents presents the information to be evaluated by agents of the other type, which should react accordingly. Therefore, agents displaying information present very diverse traits and consequently agents of the other type can also have very diverse preference lists.

The important difference between the SMP and the CFF is in how the outcome is evaluated. Instead of analysing stable configurations and of searching for algorithms that find them, the CFF proposes looking at the dynamical properties of the

population, instead. What should matter is how long marriages survive and how their duration changes when new information is presented. In particular, it is possible to define populations of interacting agents that never form long-lived matchings, despite the fact that all agents attempt to form stable pairs [3]. This is due to the presence of frustration, as illustrated in the following example.

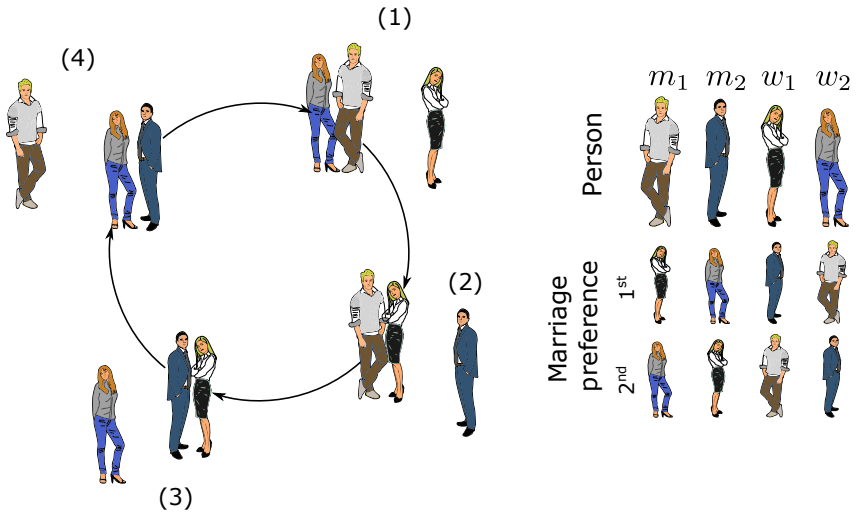


Figure 6.1: Frustrated dynamics arising in a population of (many) agents of two types, men and women, with the preferences listed on the right. In this population, if every individual mates to maximize his satisfaction (preferences), then any pair can be destabilized by unmatched individuals in the population, because in any matching there is always one individual that is completely satisfied. Cellular frustrated systems (CFSs) like this one are intrinsically unstable because even when all individuals are in stable pairs, breaking a small number of pairs is enough to destabilize the whole population and increase the number of unpaired individuals [3].

Consider a population with two types of men and women, pictured in Figure 6.1 by women (or men) dressed in casual or formal styles. Assume that men of type 1 (denoted $m_1$) prefer women of type 1 ($w_1$) to women of type 2 ($w_2$); men of type 2 ($m_2$) prefer women of type 2 ($w_2$) to women of type 1 ($w_1$), and so on as shown in the preference lists in Figure 6.1.

Assume that a man of type 1 marries a woman of type 2 (configuration (1) in Figure 6.1 ). Then, according to man's preferences, if a woman of type 1 proposes to the man in the couple, he divorces and marries the proponent woman. However, next a man of type 2 can propose to the woman in the new couple, causing a divorce and forming another couple. The cycle can go on as illustrated in Figure 6.1, and it demonstrates the effect of frustration in the population: no agent can establish a stable pair because there will always be agents that can frustrate newly formed couples. This analysis can be made more general, to consider when all agents are different and that some are initially already paired. In any case, the main conclusion

does not change: there are populations in which agents never form stable pairs [3].

Important consequences can result if major events in a population only take place when agents are matched for a minimum amount of time. This happens for certain reactions in biomolecular systems [25, 26], the cellular activation in immunology [2, 27, 28] or reproduction in evolutionary biology populations [29]). Then, it becomes clear that, despite the fact that all agents continuously interact, some will never react. This crucially depends on which agents are in the population and on the specific ordering of preferences.

Consider now that a new type of women is introduced in the population. If one assumes that different men can have different preferences towards women they never saw, then approximately one third of the men population will rank women of the new type first. If all women of the third type have the same preferences towards men, then one sixth of them will establish stable marriages. This is a meaningful fraction which shows that: i) the highly frustrated dynamics require considerable organization on preferences orderings, and ii) the dynamics can be easily disrupted by foreign elements [3].

The cellular frustration framework used these ideas to propose an alternative view on how the human adaptive immune system is activated (i.e. triggered). However, instead of men and women, there are two cell types: antigen presenting cells (APCs), and T cells. APCs present information to T cells through specialized ligands (formed by antigen bound to MHC molecules). T cells interact with these ligands with very different affinities. This information can be mapped onto a list (an interaction list or IList, similar to the preference list in the SMP) where ligands are ranked in order of decreasing affinities. T cells undergo a selection process (called the T cell repertoire education, which corresponds to a training stage). During this stage only normal (i.e., healthy) information is presented and only T cells engaging in a frustrated decision dynamics survive. Therefore, all T cells establishing long lived interactions are eliminated. This selection process establishes an ordering in T cells ILists.

An important hypothesis used by the CFF is that pairing lifetimes - i.e., the slope of the distribution of pairing lasting longer than a duration $\tau$ - are robust anomaly detection indicators. However, since accessing directly pairing lifetimes is difficult, the CFF proposes measuring the fraction of pairs lasting for a certain amount of time which is an indirect measure of the pairing lifetime. Indeed, it was hypothesized that the important function of positive selection - a training stage in the adaptive immune system that eliminates T cells that stay alone for too long - is to normalize the distribution of pairing durations so that by measuring a pairing duration, pairing lifetimes can be implicitly measured [4, 30].

Since all these concepts fit consistently in a common way of thinking of cellular

populations, this was coined as the cellular frustration framework. In the next section we detail cellular frustration algorithms for data mining applications.

## 6.4 Cellular Frustration Algorithm as an anomaly detection tool

The cellular frustration framework was created to model the immune system. As a result, most assumptions related to the behaviour of cells were inspired in the immunological reality. Even though the immune system may have evolved to perform anomaly detection accurately, it had to withstand a number of challenges and constraints that algorithms for data mining applications do not need to be concerned with. Indeed, the immune system adopted the best solutions offered by chance and natural selection, and not necessarily the best solutions that can exist. In this section we describe improvements on the application of the cellular frustration framework for anomaly detection in data mining applications. We start by defining each agent and afterwards we describe how agents interact in the several stages of the algorithm. First we describe the education stage - commonly known as training in the anomaly detection field - and discuss how it can be optimized. Afterwards we describe the detection stage, and discuss how the performance of the algorithm is evaluated.

### Agents information and Decision Rules

In the cellular frustration model considered here there are two types of agents (Figure 6.2). On one side there are $N$ presenters $P_i$ (the APCs in the immune system; $i = 1, ..., N$) and on the other side, $N$ detectors $D_i$ (the T cells).

All agents are assigned interaction lists (ILists) where the information displayed by agents of the other type is ranked. These lists play the same role as the preference lists in the SMP. All agents change pair if the information displayed by an agent of the other type is ranked higher in their ILists than the information displayed by the agent they are paired with. Furthermore, as in the SMP, all agents prefer to be paired than to be alone. Computationally, decision rules and pair formation can be written as in the Pseudo-code 6.1.

In a model with connectivity $C$, all detectors are assigned a randomly draw set of $C$ presenters they can interact with.

All data presenters present distinct information $s_i$ deriving from samples in a dataset. Many different mappings can be defined to relate sample elements $x_i$, and the information displayed by data presenters. For instance, this could even involve a dimensionality reduction pre-processing stage as it is often required to deal with

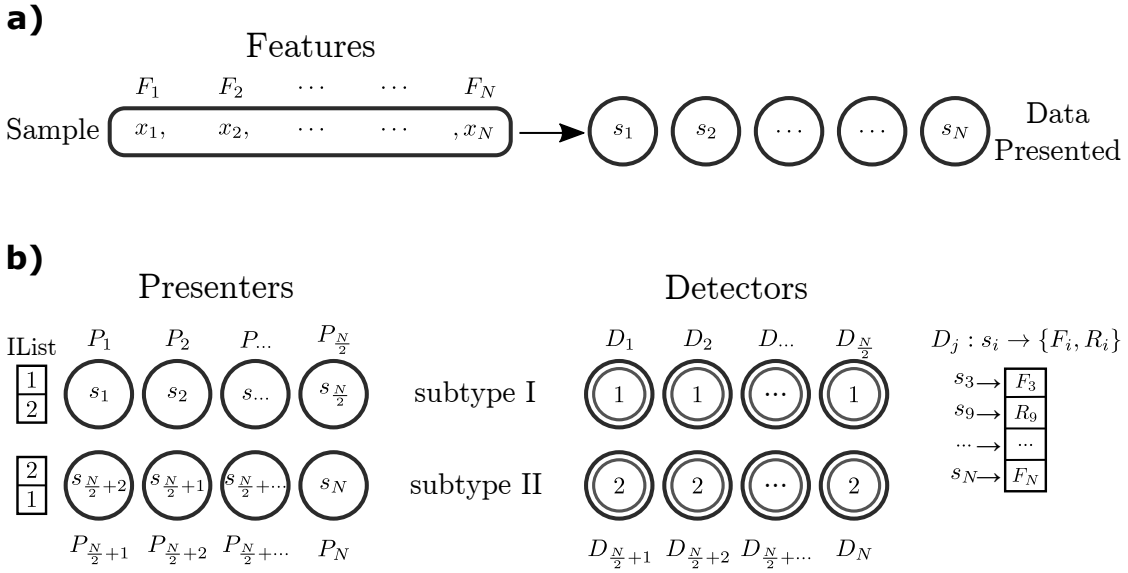


Figure 6.2: Representation of the cellular frustrated model. a) Transformation of sample values $x_i$ from $N$ features, $F_i$, onto *non-overlapping* signals $s_i$. b) Representation of the set of $N$ agents in the model. The model is composed of two types of agents, presenters $P_i$, and detectors $D_i$, with two subtypes each, with an equal number of agents. Detectors either display 1 or 2 to presenters. Presenters either prioritize interactions with detectors displaying 1 or 2. This defines the two interaction lists (ILists) shown on the left and two subtypes of presenters. Detectors, on the other hand, perceive a wider range of signals. From interactions with each presenter, detectors either perceive a $f_i$ or $r_i$ signal which derive from signals $s_i$ displayed by presenter agents. Most frequently $s_i$ signals are mapped onto a $f_i$ signal, and only rarely onto a $r_i$ signal. This mapping varies from detector to detector as described in the text. Each detector has associated ILists and will prioritize establishing pairings with agents delivering signals that ranked in the highest positions. The IList for a detector $D_j$ is shown on the right.

big samples of data. Since this was not the case in the examples discussed here, the mapping used the formula:

$$s_i = i + (x - x_{i,min})/(x_{i,max} - x_{i,min} + \epsilon) \tag{6.1}$$

where $x_{i,min}$ and $x_{i,max}$ are the minimum and maximum in the whole dataset for the $i^{th}$ feature, and $\epsilon$ is a negligible number that guarantees that different presenters display distinct information.

Detectors present only two digits, 1 or 2, and therefore they can be grouped in two subtypes $I$ and $II$. Likewise, presenters ILists rank only the two digits 1 or 2 and consequently only two types of ILists exist. This establishes also the difference between presenters of the two subtypes.

By contrast, detectors have access to a much more diverse information since the signals displayed by presenters can be continuous variables. However, to capture the

**Pseudo-code 6.1** Function establishing pairing decisions when agents $a_i$ and $a_j$, of opposite types, are put in interaction. Both agents evaluate the ranking of the signals delivered by the other agent in the pair. Here $r_{L_i}(s_j)$ denotes the rank of signal $s_j$ in agent's $a_i$ IList. $\{s_i\}$ is the set of signals displayed in a sample.

```
function DECISION({a_n}, i, j,{s_i})
    if a_i is alone ∧ a_j is alone then
        pair a_i and a_j
    else if a_i paired with a_k ∧ a_j is alone ∧
            r_{L_i}(s_j) < r_{L_i}(s_k) then
        set τ_i and τ_k to 0
        unpair a_i and a_k
        pair a_i and a_j
    else if a_j paired with a_k ∧ a_i is alone ∧
            r_{L_j}(s_i) < r_{L_j}(s_k) then
        set τ_j and τ_k to zero
        unpair a_j and a_k
        pair a_i and a_j
    else if a_i paired with a_k ∧ a_j paired with a_p ∧
            r_{L_i}(s_j) < r_{L_i}(s_k) ∧ r_{L_j}(s_i) < r_{L_j}(s_p) then
        unpair a_k and a_p
        pair a_i and a_j
        set τ_i, τ_j, τ_k and τ_p to zero
    end if
end function
```

relevant information some sort of information reduction is required. In [30] it was proposed that detectors perceive only two types of signals from each data presenter, $f$ or $r$. The $r$ signal is rarely displayed, while the $f$ signal appears frequently. Therefore, each detector establishes the following mapping for the signals read from each presenter: $s_i \rightarrow b_i$ with $b_i \in \{f_i, r_i\}, (i = 1, ..., N)$, for frequent and rare signals, respectively. Each detector reads different signals from the different presenters. Therefore, if a detector interacts with $C$ presenters, it will sense $2C$ different signals.

Several different strategies could be used to define how each detector maps sample information onto rare and frequent signals. Here each detector was assigned a different threshold probability $v_i$, drawn from a uniform distribution between 0 and $v_{max}$. Furthermore, detectors either sense rare signals on the left or on the right tail of the distribution of the sample elements displayed by a given presenter. Denoting by $F_i(s)$ the cumulative distribution estimated from training data of the elements displayed by presenter $i$, then $s_i \rightarrow r_i$ if $F_i(s_i) < v_i$ or $F_i(s_i) > 1 - v_i$, and $s_i \rightarrow f_i$ if $F_i(s_i) \geq v_i$ or $F_i(s_i) \leq v_i$, for detectors mapping sample elements on rare signals on the left or right tails, respectively. This mapping is illustrated in Figure 6.3 for the information displayed by one presenter agent.

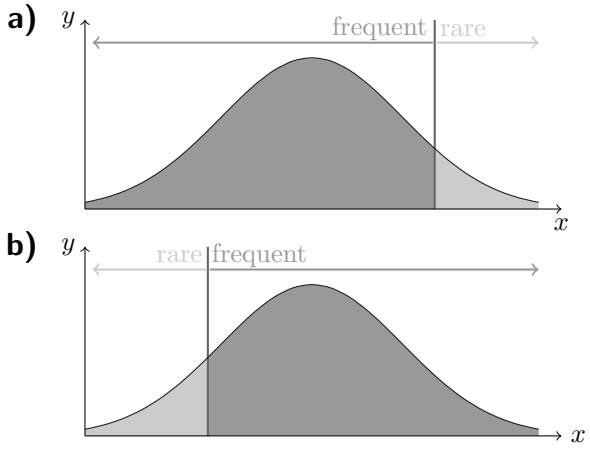


Figure 6.3: Mapping of a feature variable into frequent and rare signals. In this work detectors establish intervals mapping sample values onto rare signals either on the a) right, or on the b) left tails of the distribution function for each feature variable. The size of these intervals in the tails depends on the detector agent, and corresponds to $v_i\%$ of the events observed during education. When $v_i = 0\%$, then rare signals are not displayed during the training stage. In immunology these signals are called nonself ligands and in statistics they correspond to outliers.

Note that, as mentioned before, the way detectors map the information displayed by different detectors has an impact on the detection accuracies achieved. For instance, the detectors considered here are one-sided, since only elements on one side of the distribution tail are mapped onto rare signals. Two-sided detectors could have also been considered but we leave these and other extensions for discussion in a forthcoming publication.

## Education: main concepts

To achieve accurate anomaly detection, cellular frustrated systems (CFSs) must first undergo a training stage (also called repertoire education) during which detector ILists are changed to increasingly frustrate the overall dynamics and reach a maximally frustrated state. To understand how this guarantees accurate anomaly detection, it is important to take into consideration the mechanisms involved, thoroughly discussed in [4] and [30]. So far it has been found that CFSs can detect 3 types of anomalous patterns: 1) the presence of outliers, i.e., signals never (or rarely) displayed during education; 2) the absence of an abnormally large number of frequently displayed signals (as compared to what is observed during education); 3) the absence of combinations of signals frequently displayed during education.

Detection of these three types of anomalies rely on the organisation of ILists during repertoire education. The goal of repertoire education is to maximize

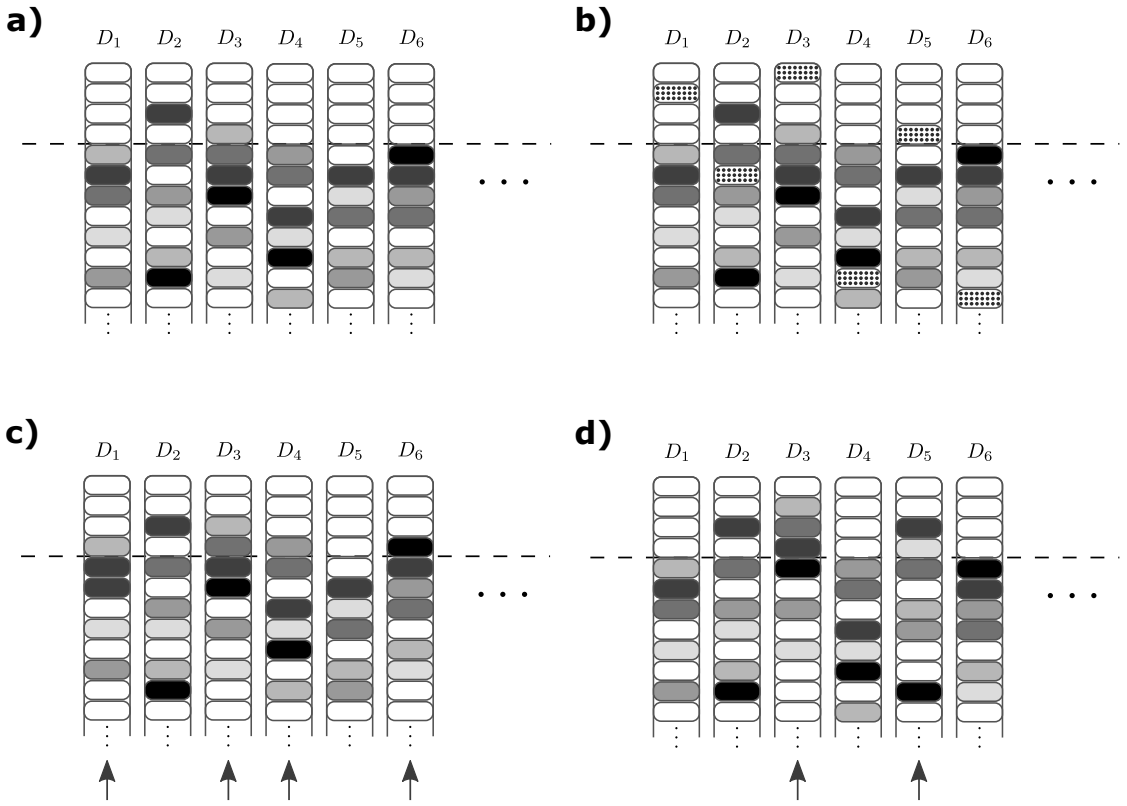


Figure 6.4: Schematic representation of the ordering in interaction lists (ILists) and their modification in the presence of anomalies. ILists for six detectors of the same subtype are represented schematically. White boxes represent signals delivered by presenters of the opposite subtype, and boxes in different grey shades represent signals delivered by different presenters of the same subtype of the detector. Only signals displayed by presenters for a given sample are represented; all others play no role in the dynamics and are omitted. a) ILists ordering when a normal sample is presented. Most detectors have on top positions - delimited by the dashed line - only signals delivered by agents of the opposite subtype. b) When a rare signal (represented in dotted boxes) that was not presented during training appears, detectors rank it in arbitrary positions; detectors $D_1$, $D_3$ and $D_5$ can establish long lived interactions. c) When the number of frequently displayed signals going absent increases beyond what is typical during training, then several detectors can have less signals delivered by presenters of the opposite subtype on top positions (detectors $D_1$,$D_3$,$D_4$ and $D_6$), which can shift the remaining signals upwards mildly but on a large number of detectors. d) Even if the number of absent frequent signals is not larger than experienced during training, detection can be triggered if the absent signals were never absent together. In that case, shifts upwards in ILists can be stronger, although may affect a smaller number of ILists (detectors $D_3$ and $D_5$).

frustration homogeneously by reducing pairing lifetimes for all detectors in the population. To avoid establishing long-lived pairings, detectors should not rank on ILists top positions signals delivered by presenters of the same subtype. Instead, on top positions there should be a set of signals frequently displayed by presenters of

the opposite subtype which can destabilize matchings with presenters of the same subtype. Therefore, after training the organisation of IList when normal samples are displayed, should be as represented in Figure 6.4a), with most detectors ILists having only signals displayed by presenters of the opposite subtype on the top. Note that in this figure, signals displayed by absent presenters are not represented, since they play no role in the dynamics.

When anomalous samples are presented, detection occurs if signals delivered by presenters of the same subtype of the detector are ranked in higher positions. This can happen either because signals have not been presented during training, in which case they will be ranked in any position in ILists (Figure 6.4b), or because frequently displayed signals become absent and detectors ranking them on top positions will push the remaining signals upwards (Figure 6.4c and d). In the last case this can happen when a number of frequently displayed signals become absent in larger numbers than happened during training. This can have a mild impact in many detectors (Figure 6.4c). The other possibility is that combinations of signals frequently displayed together, become absent. This can have a stronger impact although in a smaller number of detectors (Figure 6.4d). In practice, all three mechanisms can operate simultaneously.

## Education: algorithms

The detection mechanisms discussed above require an algorithm for ordering ILists. In [4, 30] it was proposed that the education of cells in the immune system is accomplished through a negative selection mechanism operating on the duration of pairings. Following inspiration from what is known in immunology, it was proposed that T cells (or detector agents) performing the longest pairings would be replaced by new incoming cells, with randomly ordered ILists.

Here we will show that this process can be speeded up considerably. Indeed, the immune system has a rather inefficient process of educating cells, which amounts to eliminate approximately 95% of thymocytes and replace them with new cells (with untested receptors). However, this inefficient process may be due to the fact that the immune system did not have access to mechanisms allowing edition of receptors. If one takes an artificial intelligence perspective, it is more reasonable to correct progressively ILists that led to stable pairings, instead of simply replacing them by new randomly ordered ILists. This can have the advantage of avoiding that new information destroys past experience. However, it requires designing a new strategy to order ILists.

In this article we discuss in detail a new and simple strategy. It consists in exchanging the signal that led to the longest pairings, with a randomly drawn signal

from a lower position in the IList. This strategy pushes to lower positions signals delivered by presenters of the same subtype since they produce the longest pairings. Furthermore, it can bring to top positions signals that have never (or rarely) been displayed by presenters of the same agent subtype. Indeed, it should be noted that the signal randomly drawn from a position below, is not necessarily displayed in the current sample. As a result the strategy for correcting ILists can make detection of outliers more robust than the immunological plausible strategy of replacing a detector by a new detector.

The education algorithm can then be summarized as follows (see Pseudo-code 6.2). First, detectors ILists are initialized (line 2), being assigned a set of $C$ randomly drawn presenters for interaction with each detector ($C$ stands for the detectors connectivity).

Then, the iterated frustrated dynamics is run. At each time step, a randomly drawn agent is put in interaction with an agent with signals in its IList. A new pair is formed whenever the two interacting agents prioritize this interaction (see Pseudo-code 6.1). In that case, if they were already conjugated, former pairs are terminated. The process is repeated (lines 13-17) until all agents were given a chance to chose an agent of the opposite type to interact with.

Then every detector $D_j$ involved in a pairing lasting $\tau_j$ iterations with $\tau_j > \tau_n$ undergo IList education (lines 20-31). In the Pseudo-code 6.2, the two education strategies are considered. The immunological plausible strategy (IS) simply replaces the IList by a new randomly drawn IList (line 21) while the swapping operation in the IList is considered for the artificial intelligence strategy (AIS: lines 23-27).

If after $W_\tau$ iterations (typically 10000 iterations) no agents exceeded $\tau_n$, then $\tau_n$ is updated to the largest pair duration in the last $W_\tau$ iterations (line 37). Also, every $T_S$ iterations the sample displayed by presenters is changed (lines 10-12).

Education stops when $\tau_n$ exceeds the maximum number of iterations. Then ILists are registered and added to a ILists repertoire and education of another set of ILists is restarted, but using the same connectivity assignment to each detector. Other stopping criteria could be used to finish education, like considering a maximal number of iterations.

As a side note we remark that in this work it was avoided that the two signals (frequent or rare) delivered by a presenter of the opposite agent subtype are both ranked on top positions. Indeed this would not favour detection since the absence of the frequent signal would be compensated by the presence of the rare signal. Therefore we forced rare signals delivered by agents of the opposite subtype to be ranked (and frozen) on bottom positions in ILists. This improvement in the algorithms did not change results qualitatively, and for a matter of simplification in the presentation it was omitted from the pseudo-codes.

**Pseudo-code 6.2** Repertoire education in CFAs

```
1: function RepEducation(t_max, W_τ)
2:     Initialize D_i with a random IList with f and r
3:      signals from C randomly drawn presenters
4:     Initialize {τ_i} to zero
5:     Initialize τ_n to W_τ
6:     for t in 1 to t_max do
7:         Initialize N_subs to zero
8:         Initialize τ_n^W to zero
9:         for t_w in 1 to W_τ do
10:            if t_w (mod T_S) is zero then
11:                change sample {s_i}
12:            end if
13:            for all a_i in {P_i} ∪ {D_i} do
14:                a_j: agent randomly selected from a_i
15:                     connectivity
16:                Decision({a_n}, i, j,{s_i})
17:            end for
18:            for all a_j in {D_i} do
19:                if τ_j ≥ τ_n then
20:                    if IS (immunological strategy) then
21:                        randomly permute a_j IList
22:                    else if AIS then
23:                        a_k: agent paired with a_j
24:                        p← random integer larger than
25:                             r_{L_j}(s_k)
26:                        In L_j swap content ranked at
27:                         r_{L_j}(s_k) with content ranked at p
28:                    end if
29:                    unpair a_j and set τ_j to zero
30:                    N_subs ← N_subs + 1
31:                end if
32:            end for
33:            τ_n^W ← max(τ_j, τ_n^W)
34:            Increment τ_j for all pairings
35:        end for
36:        if N_subs is 0 then
37:            τ_n ← τ_n^W, if τ_n^W < τ_n
38:        end if
39:    end for
40:    return {D_i}
41: end function
```

## Detection

Testing the anomaly detection performance of the algorithm follows closely that outlined in [30]. First it undergoes a calibration stage, to extract typical properties

**Pseudo-code 6.3** Monitoring stage of the cellular frustration algorithm.

```
1: function MONITORING(W_d, {P_i}, {D_i}, τ_A,{s_i})
2:     Initialize {τ_i} to zero
3:     Initialize c_{i,s}(τ) to zero
4:     for t_w in 1 to W_d do
5:         for all a_i in {P_i} ∪ {D_i} do
6:             a_j: agent randomly selected from a_i
7:              connectivity
8:             DECISION({a_n}, i, j,{s_i})
9:         end for
10:        for all a_j in {D_i} do
11:            if τ_j ≥ τ_A then
12:                Separate a_j from a_k and set τ_j and τ_k
13:                 to zero
14:                c_{i,s}(τ_A) ← c_{i,s}(τ_A) + 1
15:                c_{k,s}(τ_A) ← c_{k,s}(τ_A) + 1
16:                Replace a_j with a random detector
17:                 with the same connectivity
18:            end if
19:        end for
20:        Increment all τ_i
21:        Increment all c_{i,s}(τ_i)
22:    end for
23:    return {c_{i,s}}
24: end function
```

from the frustrated dynamics. In this stage agents engage in a frustrated dynamics using the decision rules in the Pseudo-code 6.1. However now, anergy is introduced, terminating pairings lasting longer than $\tau_A$ and replacing the detector involved by another detector in the repertoire with the same connectivity (Pseudo-code 6.3). In our results we used $\tau_A = 5$ iterations. During calibration only samples available for education from the normal dataset. The dynamics is run for $W_d$ iterations for each sample (typically $W_d = 10^4$ iterations). The number $c^0_{i,s}(\tau_{act})$ of long-lived pairings that lasted longer than $\tau_{act}$ iterations and involving a presenter with index $i$ when sample $s$ is presented is incremented. Defining the ordered vector $c^0_{i,(j)}(\tau_{act})$, such that $c^0_{i,(j)}(\tau_{act}) \geq c^0_{i,(j+1)}(\tau_{act}) \forall j$, then an activation threshold is established by defining $n^0_i(\tau_{act}) = c^0_{i,(x)}(\tau_{act})$ where $x = N_c \times f$, with $N_c$ the number of samples used during the calibration and $f$ is a real number between 0 and 1. Typically we use $f = 0.1$, and hence the 10% largest number of pairings lasting a time larger than $\tau_{act}$ in a sample are considered. The activation reference time was chosen to be equal to the largest pairing time during calibration, i.e., $\tau_{act} = \tau_A$.

To evaluate detection capabilities the decision dynamics is run in the detection stage in the same conditions as in the calibration stage. Presenters display either

information from $N_d^s$ samples from a self-dataset, or $N_d^{ns}$ samples from a nonself or abnormal-self dataset. Several examples are illustrated in the Numerical Results section. The CFS response to the information displayed by sample $s$ is calculated using the normalized number of pairings, $\widetilde{c}_{i,s}(\tau_{act}) = c_{i,s}(\tau_{act})/c_{i,s}(0)$, $\widetilde{n}_i(\tau_{act}) = n_i^0(\tau_{act})/n_i(0)$ according to:

$$R_s = \sum_i (\widetilde{c}_{i,s}(\tau_{act}) - \widetilde{n}_i^0(\tau_{act}))\theta(\widetilde{c}_{i,s}(\tau_{act}) - \widetilde{n}_i^0(\tau_{act})) \tag{6.2}$$

where $\theta$ is the Heaviside function. Thus the CFS response sums the increments on the number of long pairings relatively to the calibration stage, using the (normalized) number of pairings in the time interval $W_d$.

To quantify the detection accuracy we compute the true positive rate for a fixed false positive rate, $FPR$. To achieve this we create and ordered vector of population responses to the $N_d^s$ normal samples displayed in the detection stage, $R_{(i)}^s$, such that $R_{(i)}^s \geq R_{(i+1)}^s \forall i$ and find $R_x^s$, where $x = N_d^s \times FPR$. Then the true positive rate becomes $TPR = \#\{R_s^{ns} : R_s^{ns} > R_x^s\}/N_d^{ns}$, where $R_s^{ns}$ are the population responses to the $N_d^{ns}$ samples displayed with anomalies. The true positive rate is thus equal to the fraction of samples displaying anomalies with responses greater than $R_x^s$.

## 6.5 Training convergence: quantitative insights

In this section we will use a quantitative approach to understand how much faster the education strategy proposed above is, relatively to the immunologically more plausible alternative. To address this question we consider a simpler, yet similar task, capturing the essential differences between the two approaches but reducing the complexity of the problem to that of ordering a single IList.

The simpler model assumes that there are $N$ items of two types ($N/2$ from each) in a IList. By definition, it is assumed that one type of items is *correctly* ranked if it is should be ranked in top positions, . Conversely, when items of the other type appear in top positions they are *incorrectly* ranked. The aim is to find how many iterations are necessary to obtain an IList with $n$ correctly ranked items in the top $n$ positions, using two different algorithms.

The first algorithm bears inspiration from the immunological negative selection model. On each time step an item is selected from the IList. If the item is incorrectly ranked in the top $n$ positions, then a random permutation is operated on the whole IList, which corresponds to replacing the IList by a new one. This simulates the interaction of detectors with presenters producing long pairings and the subsequent negative selection of the detector.

The second algorithm reproduces the artificial intelligence education strategy, whereby selection of an incorrectly ranked item in the top $n$ positions swaps the incorrectly ranked item with a randomly selected item from the $N - n$ positions below.

The two algorithms can be modelled with the Markov models graphically represented in Figures 6.5 and 6.6. These models consist of waiting states with $m$ correctly ranked items in the top $n$ positions, $W_m$, transient education states, $E$ or $E_i$, on which the two different education strategies operate, and the absorbing state $S$ that stops the algorithm when all items are correctly ranked on the top positions.

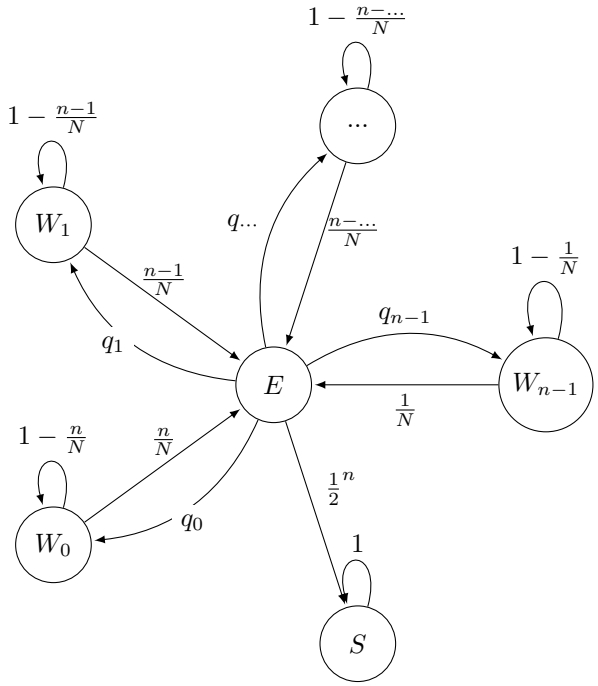


Figure 6.5: Markov chain describing the state transition that detectors undergo during education considering the immunological model. In this representation, $n$ describes the number of top positions to be corrected in a list of size $N$. The probability of transition from the $IList$ education state $E$ to the waiting states, $W$, is $q_m = \binom{n}{m}(\frac{1}{2})^{n-1}\frac{1}{2} = \binom{n}{m}(\frac{1}{2})^n$, where $m$ is the number of correct positions in the top $n$ positions in the $IList$

A fundamental difference exists between the two models. In the immunological model IList education can send the model to a $W_m$ state with any number $m$ of correctly ranked items. These states have different probabilities of sending the system to the education state $E$, which depends on the number of incorrectly ranked items. When there are $m$ correctly ranked items this probability is $q_{educ} = (n-m)/N$. If the list is sent to education, state $E$, the immunological model replaces the list by a new randomly drawn list. Therefore, from state $E$ the system goes onto a state

with $m$ of correctly ranked items with probability $q_m = \binom{n}{m}(1/2)^{(n-m)}(1/2)^m$. In particular, it reaches the absorbing state with probability $1/2^n$. Clearly, the larger $n$ the harder it takes to completely order the top positions in the list.

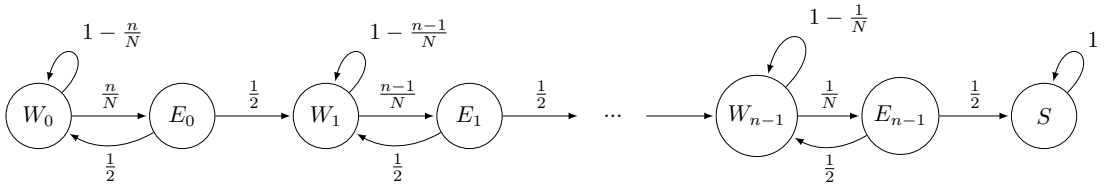


Figure 6.6: Markov chain describing the state transition that detectors undergo during education considering the proposed IList modification approach.

By contrast, in the artificial intelligence approach lists are progressively corrected. The associated Markov model has a quite different diagram as shown in Figure 6.6. In fact, each time a list enters education, which happens with the same probability as before $q_{educ} = (n-m)/N$, when it has $m$ correctly ranked items, then it either places a correctly ranked item in that position or not. Here we assume that the total number of items in the list is much larger than the number of positions to educate, $N >> n$, so that both these probabilities can be assumed to be equal to 1/2. As a result, in the artificial intelligence approach the system progresses along progressively more educated lists (states $W_m$), although it only corrects one item at each time.

These two Markov models can be described by different transition matrices, containing the probabilities of transition, $p_{ij}$, from a state $i$ to a state $j$. In the case of the immunological plausible strategy, this is:

$$P_{IS} = \begin{array}{c} \begin{array}{cccccc} E & W_0 & W_1 & \cdots & W_{n-1} & S \end{array} \\ \begin{pmatrix} 0 & \binom{n}{0}\frac{1}{2^n} & \binom{n}{1}\frac{1}{2^n} & \cdots & \binom{n}{n-1}\frac{1}{2^n} & \frac{1}{2^n} \\ \frac{n}{N} & 1-\frac{n}{N} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{N} & 0 & 0 & \cdots & 1-\frac{1}{N} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \end{array} \tag{6.3}$$

while for the case of the artificial intelligence strategy, it becomes:

$$P_{AIS} = \begin{array}{c} \begin{array}{cccccc} W_0 & E_0 & W_1 & E_1 & \cdots & S \end{array} \\ \begin{pmatrix} 1-\frac{n}{N} & \frac{n}{N} & 0 & 0 & \cdots & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \cdots & 0 \\ 0 & 0 & 1-\frac{n-1}{N} & \frac{n-1}{N} & \cdots & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \end{array} \tag{6.4}$$

To calculate the average number of steps, $K_i$, required to reach the absorbing state starting from state $i$, one considers an ensemble of lists starting in state $i$, and the ensemble of these lists in the following iteration. The average number of steps for these different configurations of lists to reach the absorbing state should differ by 1 iteration. Therefore we should have $K_i = 1 + \sum_j p_{ij} K_j$, where the sum goes over all possible states and accounts for the average number of steps required to reach the final state starting from the following configuration.

Using this equation for the immunological plausible approach it can be noted that every state $W_m$ can be written in terms of state $E$ as:

$$K_{W_m} = K_E + \frac{N}{n-m} \tag{6.5}$$

Substituting equation (6.5) in the equation for the $E$ state, we arrive at an expected number of steps to absorption of:

$$K_E = 2^n + N \sum_{j=0}^{n-1} \binom{n}{j} \frac{1}{(n-j)}, n > 1 \tag{6.6}$$

Using this solution in equation (6.5) we get the expected number of steps to absorption from a waiting state $W_m$:

$$K_{w_m} = \frac{N}{n-m} + 2^n + N \sum_{j=0}^{n-1} \binom{n}{j} \frac{1}{(n-j)}, \quad \forall m < n, n > 1 \tag{6.7}$$

Writing a general expression for the expected number of steps to absorption using the artificial intelligence strategy requires noting two conditions. First, that the expressions for the $E_m$ states can be written in terms of the expressions for the waiting states, hence:

$$K_{E_m} = 1 + \frac{1}{2} K_{W_m} + \frac{1}{2} K_{W_{m+1}} \tag{6.8}$$

Next, by using this expression in the expression for the waiting states a pattern emerges:

$$K_{W_m} = \frac{2N}{n-m} + 2 + K_{W_{m+1}} \tag{6.9}$$

Rewriting equation (6.9) in terms of the absorbing $S$ state gives:

$$K_{W_m} = 2(n-m) + 2N \sum_{j=m}^{n-1} \frac{1}{n-j}, \quad \forall\ 0 \leq m \leq n-1, n \geq 1 \tag{6.10}$$

Expressions (6.7) and (6.10) allow comparing the convergence speed for the two strategies. In Figure 6.7, it can be appreciated that the two strategies have very different

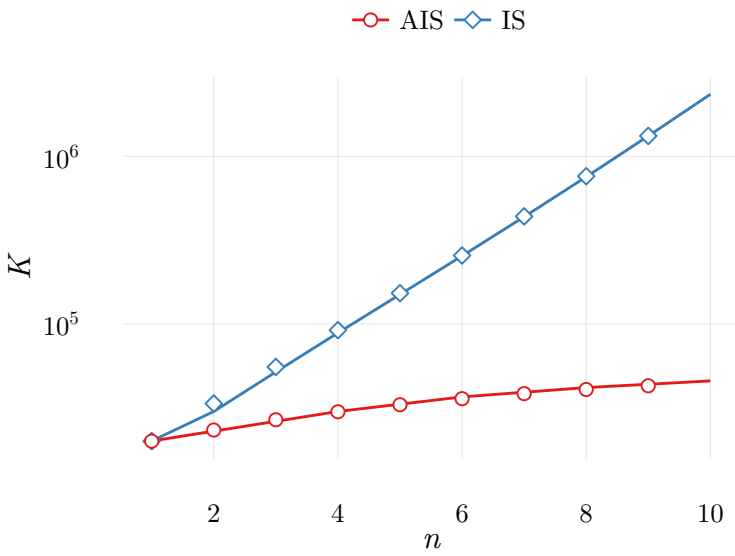


Figure 6.7: Average number of iterations required to find a list with all items in the top $n$ positions correctly ranked, for the two strategies discussed in the text: the immunologically plausible strategy (IS) and the artificial intelligence strategy (AIS).

convergence speeds even when only a small number of items has to be correctly ranked. Importantly, this difference can be of an order of magnitude.

In the next section this result is tested with the education of all ILists in a population. A fundamental difference exists, which is that all ILists have to be educated simultaneously, interfering in the education of each other.

## 6.6 Numerical Results

Here, we will use numerical results to address the two issues discussed above, namely, on the speed of convergence and on the accuracy of the new education algorithm proposed here. For these tests four different datasets were used, three from the UCI repository [31] and one available at [32]. The datasets used concern: the evaluation of wine quality [33], the well known iris dataset for species discrimination using morphological measurements [34], discrimination of two types of surfaces using scattered sonar signals (the Connectionist Bench dataset [35]) and the identification of damaged or used ball bearings (ball bearings [32]).

These datasets have samples labelled in more than one class. Hence, they are most suited for supervised classification tasks. However, for the purpose of this paper we want to evaluate our algorithm in anomaly detection. This required defining which samples belong to the normal class, presenting a sub-set of them in a training stage. In some cases, in the original dataset the number of samples in one class was too small to obtain reliable

results. In those cases groups of contiguous classes were created to define the normal and abnormal classes.

For the two studies addressed in this work - on the computational performance and on the accuracy of the new algorithm - 10 fold Monte-Carlo cross-validation was used. This amounted to randomly select 10 different normal datasets for training and detection, and running the algorithm under the same conditions.

Another important issue concerns the mapping of the information contained in samples when the number of elements was inferior to 32. In those cases, the sample was replicated an even number of times until reaching a number of presenters greater than 32. This avoids working with extremely small populations which could be blocked in stable configurations.

In the next subsections we describe the several datasets in greater detail and afterwards we will discuss the results obtained.

## Datasets

Four datasets were used in the studies. They are briefly denoted by ball bearings, iris, sonar and wines.

Table 6.1: Number of examples from each category in each test for training and detection, for the different datasets used.

| | | number of set examples | | |
|---|---|---|---|---|
| | | train | test | |
| dataset | normal training data | normal | normal | abnormal |
| ball bearings | new | 500 | 3650 | 913 |
| | worn out | 500 | 413 | 4150 |
| iris | setosa | 17 | 33 | 50 |
| | versicolour | 17 | 33 | 50 |
| | virginica | 17 | 33 | 50 |
| sonar | metal | 50 | 47 | 111 |
| | rock | 50 | 61 | 97 |
| wines | 3,4,5 | 500 | 1140 | 3258 |
| | 4,5,6 | 500 | 3318 | 1080 |
| | 5,6,7 | 500 | 4035 | 363 |
| | 6,7,8 | 500 | 2753 | 1645 |
| | 7,8,9 | 500 | 560 | 3838 |

The ball bearings dataset [32] derives from Fast Fourier transforms (fft) of acceleration time series signals in essays with new or worn out (broken, damaged or even used) ball

bearings. There are $N_f = 32$ features and 4150 samples deriving from essays with new ball bearings and 913 from worn out ball bearings. Training for anomaly detection tests used 500 samples from either, new or worn out ball bearings samples (Table 6.1).

The iris dataset was introduced by R. A. Fisher and is probably the most widely known dataset in the pattern recognition literature. This dataset comprises 50 samples describing three types of iris flowers by their width and length of petal and sepal ($N_f = 4$). Anomaly detection tests used a subset of samples from either one of the three classes for training, while examples from the other flower types were considered anomalous (Table 6.1).

The sonar dataset was collected by T. Sejnowski and R. Paul Gorman for discerning two types of surfaces using scattered sonar signals. The two surfaces considered were a roughly cylindrical rock and a metal cylinder. Several examples have been collected for the two surfaces at different angles and conditions. Overall signals have $N_f = 60$ features capturing information from reflected ultra-sounds and there are 97 samples from rock surfaces and 111 samples from metal surfaces. Again, tests considered that either type of material could work as the normal dataset.

Finally, in the wine dataset 4898 white wines are characterized in terms of $N_f = 11$ chemical-physico properties, such as pH, alcohol, fixed or volatile acidity, etc.. A quality score from wine tasting evaluation is also provided. In practice scores from 3 to 9 have been awarded, 3 corresponding to a very bad wine, while 9 is awarded to wines of astounding quality. The aim of this dataset is to predict wine quality based only on physiochemical properties.

The number of wines scored with each score varies considerably. Wines evaluated with scores 3 and 9 are only a few: 20 and 5 respectively. Likewise, wines evaluated with scores 4 and 8 represent only a small fraction ($\sim 3\%$ each) of the total. Finally, wines evaluated with scores 5, 6 and 7 appear respectively 30%, 45% and 18% of the times.

To evaluate the anomaly detection algorithm it was necessary to define which sub-set of wines defined the normal class. To avoid having normal classes with too few examples, groups were defined with wines scoring 3,4 and 5, or 4,5 and 6, etc. (see Table 6.1). It was then possible to define sub-sets of 500 wines for training, and use the remaining for testing (Table 6.1).

## Convergence Tests

The first numerical results reported here concern the speed of convergence of the new AIS education algorithm as compared with the immunologically plausible strategy. In Table 6.2 the average number of iterations required to reduce all pairing durations below 180 iterations are shown.

In all experiments, the AIS converged substantially faster by at least an order of magnitude. It can also be remarked that some datasets were more difficult to train than others which appears to be consistent in the two training strategies. For instance, the sonar dataset requires typically more iterations.

Table 6.2: Average number of iterations required to reduce all pairing durations below 180 iterations during $W_\tau$ iterations (results in millions of iterations).

| dataset | normal training data | education strategy | |
|---|---|---|---|
| | | AIS | IS |
| ball bearings | new | $0.5 \pm 0.07$ | $8 \pm 5$ |
| | worn out | $0.5 \pm 0.1$ | $6 \pm 4$ |
| iris | setosa | $0.5 \pm 0.4$ | $7 \pm 4$ |
| | versicolour | $0.5 \pm 0.06$ | $6 \pm 4$ |
| | virginica | $0.5 \pm 0.1$ | $7 \pm 4$ |
| sonar | metal | $1.3 \pm 0.4$ | $13 \pm 5$ |
| | rock | $1.4 \pm 0.4$ | $15 \pm 4$ |
| wines | 3,4,5 | $0.7 \pm 0.2$ | $11 \pm 5$ |
| | 4,5,6 | $0.6 \pm 0.2$ | $10 \pm 5$ |
| | 5,6,7 | $0.7 \pm 0.3$ | $11 \pm 5$ |
| | 6,7,8 | $0.7 \pm 0.3$ | $9 \pm 4$ |
| | 7,8,9 | $0.6 \pm 0.2$ | $10 \pm 5$ |

In these results the target value of 180 iterations was chosen because it corresponded to a pairing duration that could be attained within an acceptable computational time. To complement these results, in Figure 6.8 the number of iterations required to have all agents pairing durations below $\tau_n$ is plotted. These results are an indirect measure of the IList organisation, i.e., of the number of educated positions as analysed in Figure 6.7. Results in Figure 6.8 considered populations trained with the wine dataset, when the normal training data had wines with quality scores between 5 and 7. These results represent the typical behaviour of $K(\tau_n)$, also observed in the other systems.

These results show that the immunological strategy requires a number of iterations that grows faster (faster than exponentially) than the artificial immune strategy for an equivalent level of IList organisation. Therefore, these results agree qualitatively with those described by the simplified model for the education of a single IList.

## Anomaly detection performance

To compare the precision of the new training strategy with the immunologically more plausible strategy, ROC curves for anomaly detection tests with the several datasets were obtained (Figure 6.9). Furthermore, a comparison with the one-class support vector machines is also provided. To establish a fair comparison among the several methods, all algorithms used the same samples for training and testing. Furthermore, since the aim is also to evaluate the robustness of different algorithms, the normal class used for training

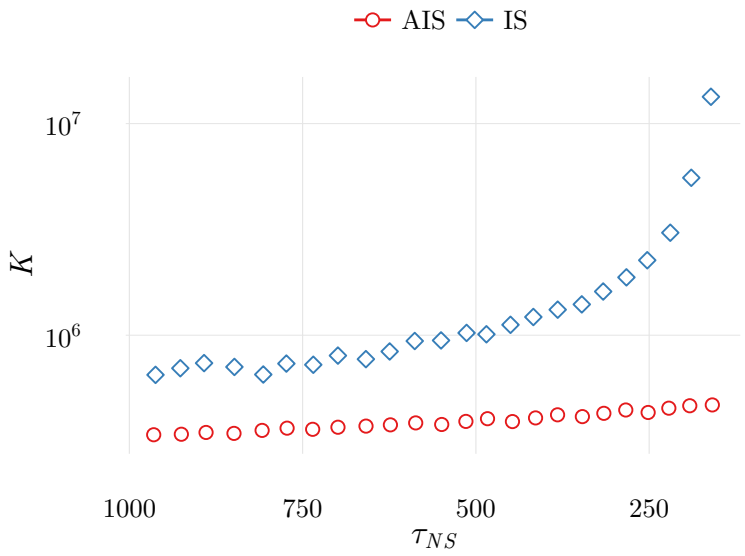


Figure 6.8: Average number of iterations as a function of the pairing duration threshold $\tau_n$ reached.

was chosen by selecting sub-sets with the different classes in each dataset (see Table 6.1). All the results presented next used a same fixed set of parameters and, in the case of the SVM, the same kernel.

The particular choice of parameters can be critical and therefore, needs to be discussed. This discussion is not always easy to make with total fairness since, in semi-supervised anomaly detection only information from a single class is available. As a result, for any new method the developer tests countless variations and incorporates his knowledge in selecting standard parameters. Certainly, only with a growing number of studies and on different datasets will it be possible to establish definite conclusions on how different algorithms compare.

The parameters used in CFSs were relatively simple to establish and are listed in Table 6.3. This seems a long list, however results do not critically depend on most of them. In many cases, their choice follows naturally from the detection mechanisms identified in [4, 30].

For instance, the threshold probability $v_{max}$ should be small but nonzero, to allow discrimination of outliers and of abnormal samples. Of course that the best value for $v_{max}$ should depend on the dataset because, if only outliers are to be found then $v_{max}$ should be zero. On the other side, if no outliers exist, then detectors using $v_{max} = 0$ only participate in frustrating the dynamics. In the supplementary materials 1 (SM1) the average $TPR$ obtained with a $FPR$ of 10% is shown for the different datasets and for $v_{max} = 0, 5, 10\%$. These results show that detection can vary with $v_{max}$. This is clear in the results obtained with the ball bearings and the wines datasets. From these results it seems clear that $v_{max} = 5\%$ is generally a good compromise.

Table 6.3: Parameters used in cellular frustration algorithms.

| | |
|---|---|
| Threshold Probability $v_{max}(\%)$ | 5 |
| Number of populations in the repertoire | 12 |
| Detectors connectivity, $C$ | 20 |
| Education Window $W_\tau$ (iterations) | $10^4$ |
| Education time sampling window $T_S$ (iterations) | 100 |
| Detection window $W_d$ (iterations) | $10^4$ |
| Anergy time $\tau_A$(iterations) | 5 |
| Detection pairing duration to activate response $\tau_{act}$ (iterations) | $\tau_A$ |
| Calibration parameter $f$ | 0.1 |

In [4] it was shown that a repertoire composed of several independently educated sets of detectors, could improve detection rates, when a single outlier was presented. This happens because the number of ILists that can rank outliers on top positions is increased. The results shown in SM2 capture some improvement when the number of populations in the repertoire increases from 1 to 12. However, the number of populations in the repertoire does not seem to have a critical impact in anomaly detection rates. This conclusion is valid as far as the current datasets are concerned. It is always possible that in other datasets the most frequent anomaly would correspond to the appearance of a single outlier. Then, the influence of the number of populations in the results could be important [4]. This can be particularly relevant in the context of intrusion detection, because attackers try to explore such vulnerabilities. In the remaining results presented next we choose repertoires with 12 populations.

In [4, 30] it was shown that detector's connectivity - the number of presenters a detector can interact with - could have an impact in anomaly detection performances and also on training convergence. To understand this it should be recalled that, using the plausible immunological training strategy, only a few top positions (typically not larger than 10; see results in section 6.5) will be ordered. That is, on top positions in ILists there will be mostly signals delivered by presenters of the opposite subtype. In the following positions, ILists are relatively disordered, with signals delivered by presenters of both subtypes. As a result, in populations with large connectivities, the probability that a detector interacts with signals in the ordered region, is small. Consequently detection performances tend to be poorer. Furthermore, education also requires more time to reduce $\tau_n$. On the opposite extreme, for very small connectivities, fluctuations in the number of

signals present in a sample and ranked on ILists top positions increase. This also leads to a less organized dynamics.

Two types of results confirm these analyses. First in SM3 it is shown that, for the immunological strategy the number of iterations required to reach a given maximal pairing duration $\tau_n$, has a minimal value for intermediate connectivities. Interestingly convergence of the artificial intelligence algorithm became much more insensitive to connectivity changes. In what concerns the impact of connectivity on the anomaly detection accuracies, results are much less clear for both strategies, and this is likely to be due to the relatively small number of independent features present in the datasets used. However, in the immunological plausible strategy there are datasets - for instance, the ball bearings dataset - in which the largest connectivities can produce clearly poorer results. In some cases, however, results are not very sensible to changes in connectivity, as happens for instance, with the sonar dataset. In any case, and interestingly, anomaly detection performances of the new education strategy are almost insensitive to connectivity changes for the studied datasets (see SM4) except if connectivity is extremely small. This, we believe, is due to the improved ordering in ILists. This result is important because it reduces the number of parameters to tune. Therefore, as a general conclusion, the connectivity should be chosen to take moderate values, within the range of a few dozens, specially for computational convenience reasons.

In order to gain good generalization capabilities, it was shown that the time sampling window $T_S$ should be small [30] and the education window $W_\tau$, used to decrease $\tau_n$, should be large - i.e., $W_\tau/T_S \sim 100$ - to correct detectors only depending on their performances in a large number of samples. The results we obtained (see SM5) do not exhibit such a dramatic effect as the one reported previously [30]. In some cases, can even seem to contradict these previous results ( as in the iris dataset, with virginica as normal class, or in the sonar dataset with metal as normal class), although we believe that this can be due to the small number of samples in these examples. More interesting, is the robustness demonstrated by the new AIS strategy to variations in $W_\tau/T_S$. This is interesting because again it shows that results became independent of the choice of these parameters.

Next, with respect to the detection window $W_d$, this was chosen to be $10^4$ because one needs good statistics to establish pairing lifetimes. However, as can be appreciated in SM6, increasing this value further does not further improve results.

The anergy time $\tau_A$ was chosen having in mind that the distribution of pairing durations decays exponentially. Therefore the occurrence of pairings lastings longer than typical pairing lifetimes may not provide additional information. On the contrary, using small values for $\tau_A$ improves statistical accuracy since more pairings can be tested. In fact, since detectors minimum pairing lifetime is of the order of 5, the number of pairings lasting longer than this value can represent 40% of the total number of pairings. Therefore, $\tau_A \simeq 5$ - the value used in [30], seems an acceptable choice. However, values up to $\tau_A \simeq 20$ would produce similar, if not slightly better results (see SM7). Finally, we should mention that $\tau_A$ should be always larger or equal to 2 because otherwise generalized kinetic proofreading

would not take place. However, we should note that in a single iteration there are agents that are selected by more than 10 agents for interaction. Therefore, even for small $\tau_A$ values, kinetic proofreading is already deeply present.

Finally, the calibration parameter $f$ was chosen to be 0.1. However, its impact in the anomaly detection performance of the algorithm is also reduced provided $f$ is not too small (see SM8). The $f$ parameter was first introduced in [36] to take into account knowledge of the typical pairing durations observed in the calibration stage. Since detection mechanisms involve the number of long lived pairings, it could be expected that only those agents performing the longest pairings should be considered. The results we present in SM8, show that if $f < 0.05$, performances deteriorate. This can be due to the fact that not enough agents that play an important role in the discrimination are participating. Therefore, $f$ should take larger values.

The interesting result if that if $f$ takes maximum values the results are almost not changed. This suggests that the calibration stage could be eliminated, which represents an important simplification in the algorithm. However, it is not clear to us how general this conclusion may be, especially having in mind future developments of the algorithm. This was the reason why the calibration stage was kept in this work.

To conclude, while there are several parameters at play, whose values have to be defined, selection of reasonable values is not difficult to establish following our understanding of the detection mechanisms. Consequently, the results presented in Figure 6.9 are robust relatively to their variation.

In contrast, the choice of the kernel in one-class SVM influences considerably the results. For the results presented in Figure 6.9, we chose the kernel that gave better overall results (a polynomial kernel with degree 2 and $c = 0, \nu = 1/N_f$ [37]). A comparison with results obtained with other kernels can be found in SM9.

The set of results in Figure 6.9 allow drawing two main conclusions by analysing the $TPR$ at a 10% $FPR$ on the several plots. First, the artificial intelligence algorithm proposed here has similar precision to the more immunological plausible alternative. Therefore, the new algorithm is interesting especially because it increases training speed by one-fold, at least.

The other important result is the comparison with the one-class SVM algorithm. The SVM in some cases is more precise - for instance, in the ball bearings dataset, when the normal class consists of samples obtained from new ball bearings, or, in the sonar dataset, when the normal class consists of data arising from sonar signals reflected by rock. However, on both cases, when the normal class is formed by samples from the other class, detection is not achieved at all. This suggests that at least, CFSs present more robust results. Of course that it may be argued that SVM methods require a judicious choice of the kernel in each case. This however, is problematic in many applications especially when semi-supervised anomaly detection is required. In the SM9 results obtained with other kernels are also presented, demonstrating overall poorer performances.

It should be mentioned that the ROC curves presented in Figure 6.9 result from a 10

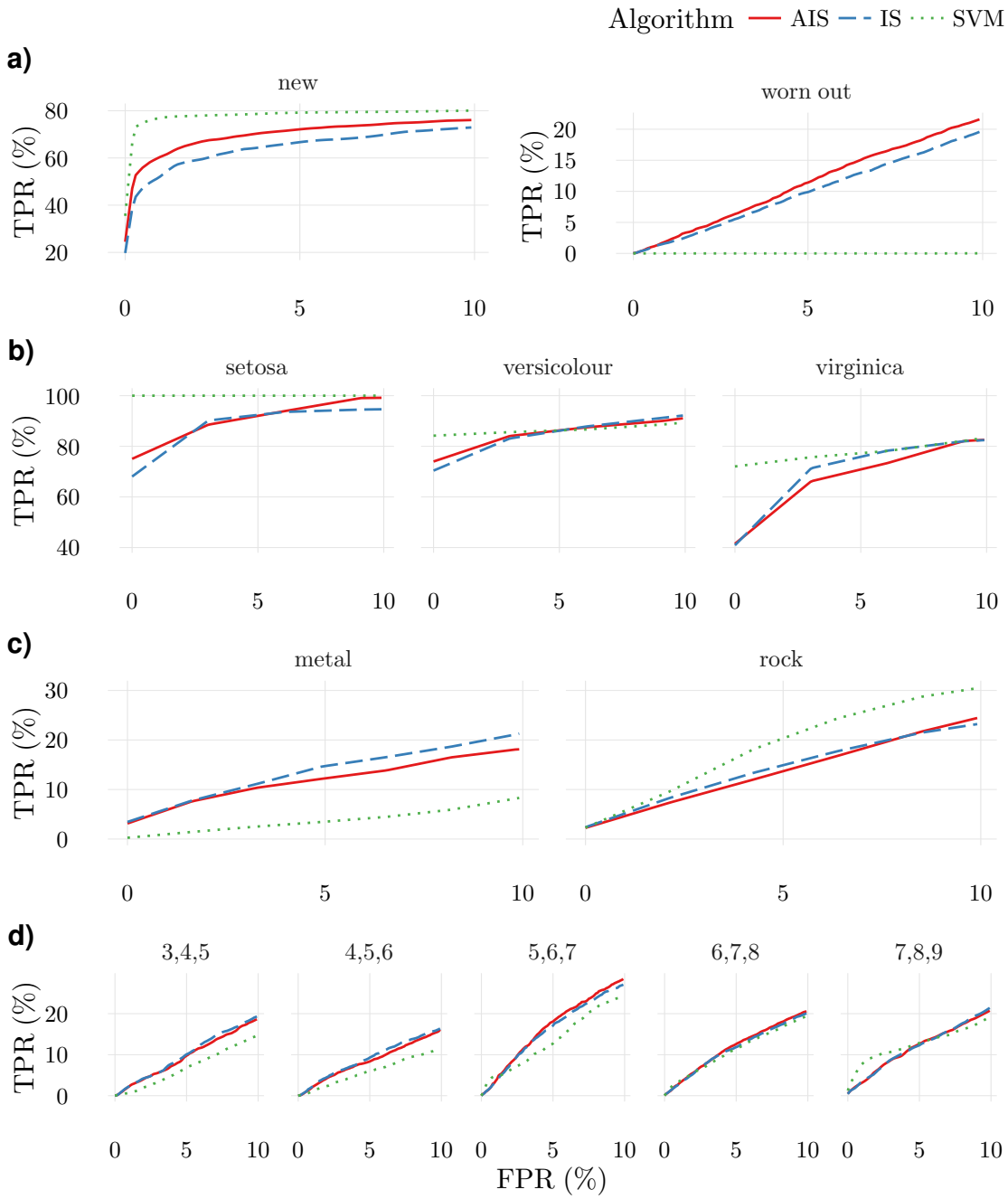


Figure 6.9: Average ROC curves obtained from 10 fold Monte-Carlo cross-validation, for the several datasets and for the several normal class sub-sets defined in Table 6.1 and mentioned in each plot header. Shown are results from the two cellular frustrated algorithms (CFAs) (parameters as in Table 6.3) and the one-class SVM with the kernel producing the best results. The quantitative value of the true positive detection rate (TPR) at at a 10% false positive rate (FPR) as well as its standard deviation can be found in SM2. Overall these results show that cellular frustrated algorithms achieve comparable, if not better precisions than one class SVMs.

fold Monte-Carlo cross-validation. Variability in these experiments exists but it is fairly similar among the two cellular frustrated algorithms, as can be appreciated in SM2. SVMs have smaller variabilities, which can be expected given the stochastic nature of CFAs.

Table 6.4: $TPR$ for 10% $FPR$ for the two strategies (AIS and IS) when $v_{max} = 0\%$ and $v_{max} = 5\%$. On the last three columns are the results using a simple rule using simply the number of rare signals mapped from each sample, as explained in the text.

| | | | AIS | | IS | | #rare signals | | #rare sig. in ILists |
|---|---|---|---|---|---|---|---|---|---|
| test | dataset | normal training data | 0 % | $[0,5]$ % | 0 % | $[0,5]$ % | 0 % | 5 % | $[0,5]$ % |
| 1 | ball bearings | new | 79.8 | 76.1 | 79.8 | 74.5 | 80.6 | 79.6 | 79.6 |
| 2 | | worn out | 10.4 | 22.0 | 10.4 | 19.7 | 8.0 | 12.1 | 13.4 |
| 3 | iris | setosa | 98.8 | 99.5 | 95.6 | 96.4 | 99.4 | 99.4 | 100.0 |
| 4 | | versicolour | 90.9 | 89.8 | 90.1 | 92.5 | 90.6 | 90.6 | 90.3 |
| 5 | | virginica | 82.5 | 82.3 | 84.4 | 81.5 | 74.5 | 74.5 | 78.4 |
| 6 | sonar | metal | 19.5 | 17.4 | 17.7 | 20.9 | 10.3 | 12.4 | 19.3 |
| 7 | | rock | 22.3 | 25.9 | 23.2 | 23.4 | 29.3 | 29.2 | 26.9 |
| 8 | wines | 3,4,5 | 12.4 | 18.5 | 12.5 | 19.6 | 13.8 | 14.2 | 17.6 |
| 9 | | 4,5,6 | 11.8 | 16.5 | 11.8 | 16.3 | 12.4 | 13.2 | 15.6 |
| 10 | | 5,6,7 | 15.8 | 28.2 | 15.8 | 26.9 | 16.7 | 26.1 | 28.1 |
| 11 | | 6,7,8 | 13.4 | 20.7 | 13.4 | 20.1 | 14.2 | 20.0 | 20.1 |
| 12 | | 7,8,9 | 18.7 | 20.7 | 18.7 | 20.9 | 21.2 | 17.0 | 21.1 |

Detection in CFAs can arise from two types of mechanisms: detection of outliers or detection of an increased number of absent frequently displayed signals. The two mechanisms can take place simultaneously, and consequently except in special cases (as those discussed in [30]), it is not always easy to clearly point which mechanism is playing a crucial role. In order to enlighten this point with respect to the present datasets, Table 6.4 compares the performance of CFAs with $v_{max} = 0\%$ and $v_{max} = 5\%$ and with results deriving from two methods based on simple rules. These two methods simply count the number of rare signals appearing in each sample in the detection stage and establish the TPR as the fraction of anomalous samples having a number of rare signals larger than found in 90% of the normal samples.

The two methods based on simple rules differ on how sample elements (i.e., features) are mapped onto rare signals. In the first method (columns 8 and 9 in Table 6.4) an

element in a sample is mapped onto a rare signal if it lies in a tail (either, left or right tail) of the corresponding feature distribution. Only data used during education is used to estimate the tail region. Therefore, for 0% tails (column 8 in Table 6.4), only sample features outside the range of values observed during education produce rare signals. The second method (results in the last column in Table 6.4) counts the number of rare signals in the detectors ILists used in the CFAs with results listed in columns 5 and 7 of Table 6.4).

Analysing Table 6.4 it is possible to conclude that:

1. in some tests, detection of outliers is responsible for the anomaly detection. This happens in tests $1, 3, 4, 7, 12$, for which the simple rule counting the number of outliers in samples (the number of rare signals in 0% tails) produces similar TPRs than CFAs with $v_{max} = 0\%$.

2. detection in tests $6, 10, 11, 12$ can be explained as resulting from the presence of a larger number of features with values in the tails than typically happens in normal samples, since the number of rare signals in ILists is enough to explain CFAs results with $v_{max} = 5\%$. Still, it should be noted that tails of different sizes must be considered and it would not be enough to consider a single tail with 5% of the values. Therefore, even if a simple rule could be devised, it requires already some computational complexity.

3. test 2, and to a lesser extent, tests 5 and 8, indicate detection of correlations in the absence of frequent signals.

In general terms, one can conclude that, although the majority of datasets may not require algorithms as elaborate as CFAs to achieve results with the accuracies reported here, it is clear that this cannot be known in advance, and also that some tests demonstrate the need for using this type of algorithms. Indeed, test number 2 clearly demonstrates that this class of algorithms is needed to perform accurate anomaly detection.

## 6.7 Conclusions

The cellular frustration framework showed a new way of looking into cellular interactions in the adaptive immune system and how they could work to produce an effective surveillance system. In particular, in a recent work we showed that cellular frustrated systems could be used to perform location statistical tests with performances that could outperform well known statistical tests, like the t-test or the KS-test [30]. In that work, using synthetic data we also showed that CFSs could compete with support vector machines.

The goal of this work was two folded. On one side we wanted to test cellular frustration algorithms using real datasets. On the other side we wanted to understand if simpler versions of the cellular frustration algorithm could be devised to produce similar, if not better results. These improved algorithms would not have to follow the immunological reality closely, taking instead a more general artificial intelligence approach. Therefore, in

this work, in the training stage, instead of replacing detectors establishing the most stable pairings by new agents, small corrections were introduced in their ILists to incorporate this new knowledge. The new algorithm proved to be at least one-fold more efficient in computational terms, and anomaly detection rates remained equivalent to the ones obtained with the more immunological version of the algorithm. Furthermore, the new algorithm also gained robustness, since it was found that anomaly detection rates only depended on a single parameter (within the reasonable ranges of variation of the parameters). This robustness improvement can also increase by one extra fold the computational efficiency of the algorithm since it reduces the size of the detectors repertoire used.

Therefore, the algorithm proposed here reduced the complexity in initial proposals [4, 30, 36] by eliminating the need of using the calibration stage and by reducing the number of parameters that one should tune to only one. It should be mentioned, however, that these conclusions are restricted to semi-supervised anomaly detection applications with stationary data. It is possible that in dynamic contexts or in the adaptation of the algorithm for classification tasks, some of these conclusions do not apply.

In this work we also compared CFAs with SVMs. It was found that the choice of the kernel in SVMs could be critical when general semi-supervised anomaly detection applications are to be considered. In this respect, CFAs are considerably better than one-class SVMs because when little is known about the type of anomalies appearing, an incorrect kernel choice can easily lead to no detection, while CFAs still perform reasonably well. On the other side, it should be mentioned that SVMs have the advantage of being considerably faster than CFAs (by almost two orders of magnitude) when datasets have a small number of samples and a small number of features. For large datasets CFAs can be competitive, although we leave investigation on this issue for future work.

To sum up, the important conclusions that can be drawn from this work is that CFAs can be competent data mining algorithms and that, given that CFAs explore a wide range of detection mechanisms poorly studied so far, an enormous range of future improvements can still be developed. These can take inspiration from the immune system, as well as ideas from the artificial intelligence field.

## 6.8 Supplementary Materials

### SM1-Impact on results of variations of $v_{max}$

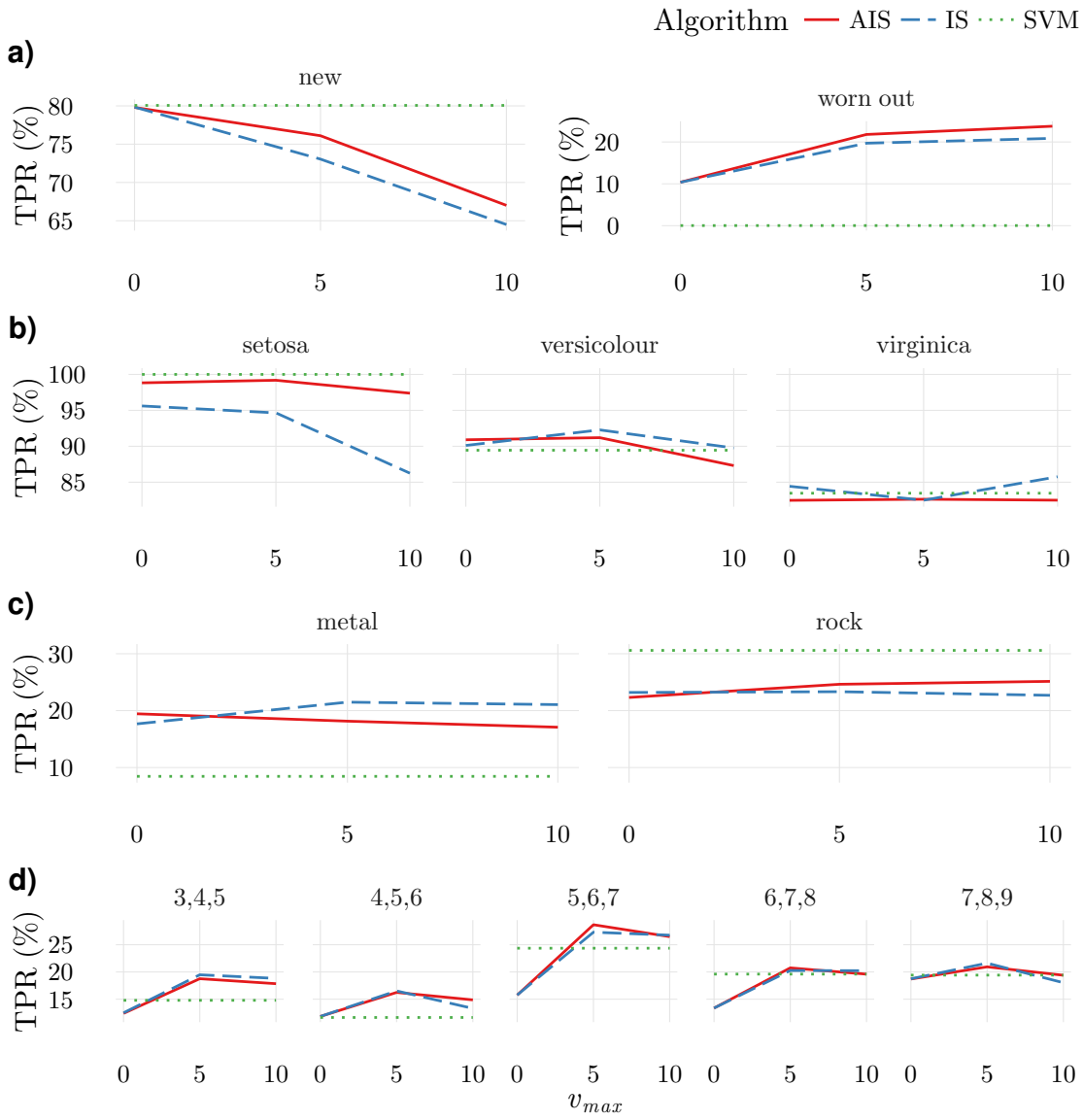


Figure 6.10: Average true positive rates (TPR) obtained when the maximal probability of perceiving the information displayed by a presenter as a rare signal is $v_{max}$ for the different strategies (AIS and IS) and on the different datasets. By order of appearance, a), b), c), d), describe the results for the ball bearings, iris, sonar and wine quality datasets, respectively. The title of each plot represents the dataset class used as the normal class, while the remaining samples were considered as abnormal. All results were obtained after 10 executions, each with an initial different set of samples. Results refer to a 10% $FPR$. Overall discrimination is best for $v_{max} > 0$ which means that discrimination is not exclusively due to the presence of outliers.

## SM2-Average $TPR$ obtained for the several algorithms

Table 6.5: Average $TPR$ and associated standard deviations obtained considering a 10% $FPR$. Also shown is the impact of increasing the repertoire population size from 1 to 12 populations of detectors in CFAs. Here, $N.D.$ stands for "no detection". Overall, these results show that: 1)larger repertoires improve slightly the accuracy; 2) CFAs have comparable, if not better overall accuracies than one class SVM.

| | | AIS | | | | IS | | | | SVM | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Number of populations | | 1 | | 12 | | 1 | | 12 | | | |
| dataset | normal training data | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| ball bearings | new | 75.8 | 7.4 | 76.1 | 6.3 | 71.6 | 9.7 | 74.5 | 8.7 | 80.1 | 0.2 |
| | worn out | 20.1 | 5.5 | 22.0 | 8.1 | 18.0 | 3.9 | 19.7 | 6.9 | $N.D.$ | $N.D.$ |
| iris | setosa | 94.1 | 13.6 | 99.5 | 1.7 | 94.2 | 13.5 | 96.4 | 7.2 | 100.0 | 0.0 |
| | versicolour | 90.0 | 10.4 | 89.8 | 9.9 | 88.1 | 13.6 | 92.5 | 9.5 | 89.5 | 4.5 |
| | virginica | 83.6 | 6.3 | 82.3 | 9.5 | 80.8 | 11.2 | 81.5 | 9.2 | 83.5 | 7.5 |
| sonar | metal | 17.4 | 11.6 | 17.4 | 9.7 | 18.1 | 12.6 | 20.9 | 9.3 | $N.D$ | $N.D$ |
| | rock | 24.9 | 6.8 | 25.9 | 5.8 | 24.3 | 5.9 | 23.4 | 8.0 | 30.6 | 3.7 |
| wines | 3,4,5 | 16.5 | 4.2 | 18.5 | 3.4 | 17.9 | 4.6 | 19.6 | 2.0 | 14.8 | 1.5 |
| | 4,5,6 | 15.3 | 2.9 | 16.5 | 2.3 | 15.9 | 3.3 | 16.3 | 2.4 | 11.6 | 1.2 |
| | 5,6,7 | 27.4 | 1.7 | 28.2 | 1.1 | 25.1 | 3.2 | 26.9 | 2.4 | 24.4 | 1.0 |
| | 6,7,8 | 20.2 | 2.2 | 20.7 | 1.5 | 18.9 | 2.7 | 20.1 | 2.0 | 19.6 | 0.8 |
| | 7,8,9 | 19.7 | 2.2 | 20.7 | 3.37 | 21.6 | 2.63 | 21.0 | 2.4 | 19.4 | 1.1 |

## SM3-Impact of varying connectivity $C$ on $\tau_n$

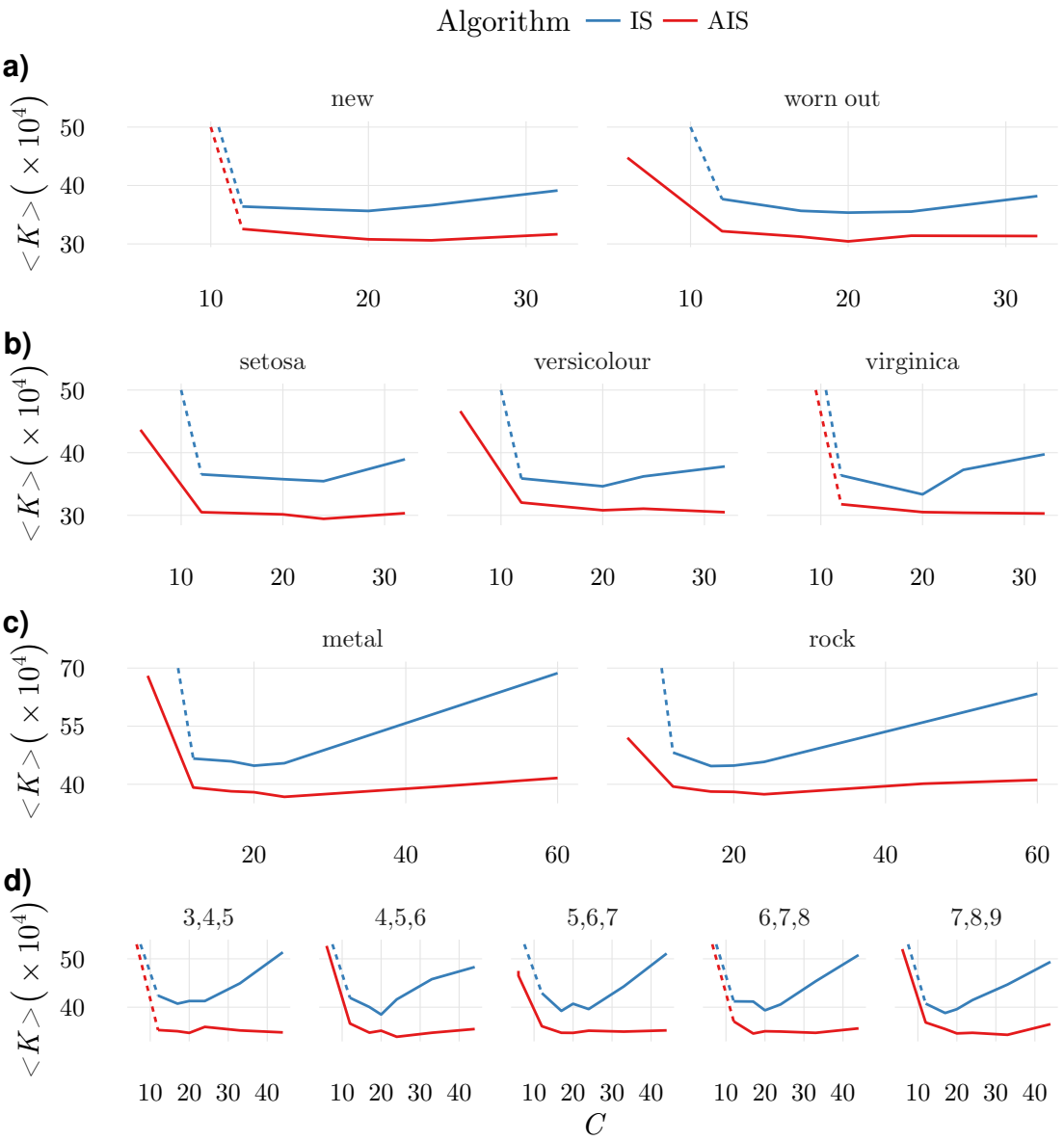


Figure 6.11: Average number of iterations required to reach the fixed maximal pairing duration $\tau_n = 560$ during education, for 10 different populations. These results show that for the immunologically plausible algorithm, smaller connectivities can considerably improve the convergence speed. This effect is almost absent in the new artificial intelligence algorithm, which highlights the good convergence behaviour of the new algorithm even when large connectivities are considered. When the connectivity is too small - the smallest connectivity considered was $C = 6$ in all cases - convergence slows considerably, and in some cases repertoire education did not converge within the maximal number of iterations attended ($20 \times 10^6$ iterations). In the later cases, dashed lines are used to serve as guides to the eye, to illustrate a steep increase in the number of iterations required for convergence.

## SM4-Impact on results of varying detector's connectivity

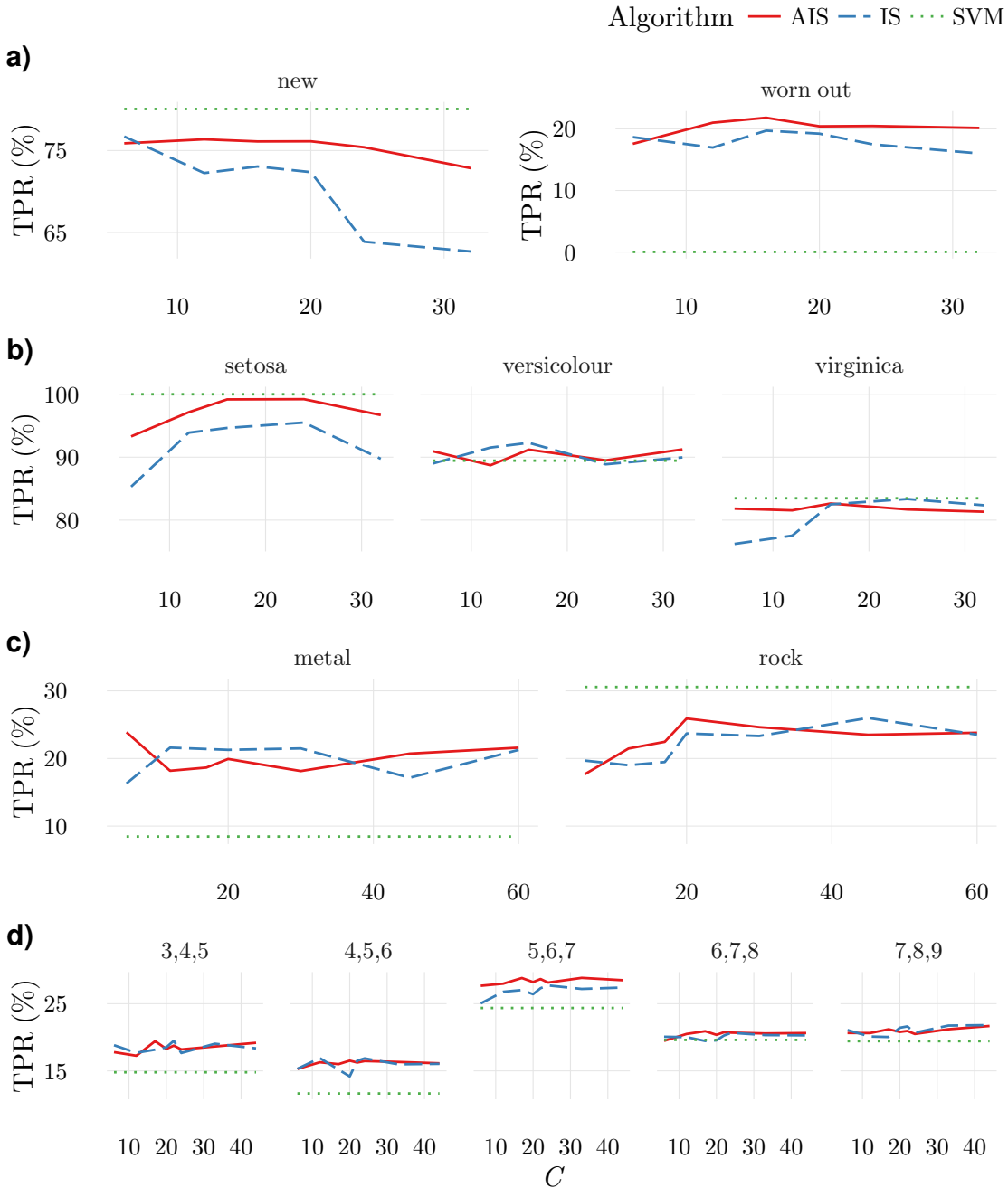


Figure 6.12: Average $TPR$ obtained for different values of detectors connectivity, $C$. Overall increasing detectors connectivity should decrease discrimination capability. This is true for the IS and for systems with a large number of features which are also harder to educate. For the datasets analysed here the impact of $C$ is only mild since the number of features in the several cases is relatively small. In any case, it seems clear that $C$ should not be too small (in this case education can be very hard; see SM3) and there can be an advantage in not having full connectivity.

## SM5-Impact on results of varying $T_S$

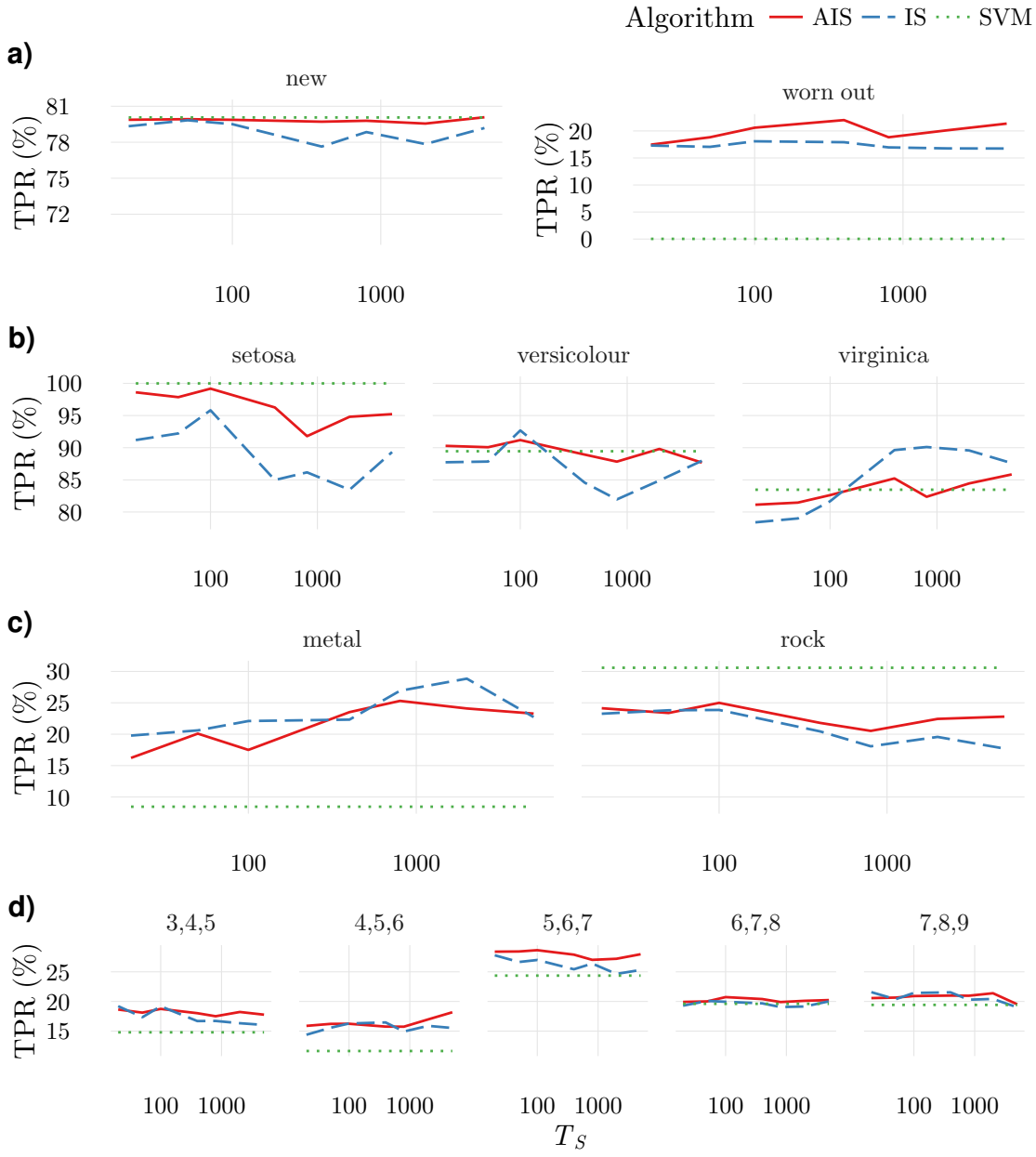


Figure 6.13: Impact of changing $T_S$, the number of iterations separating the presentation of different samples during training, when the time interval $W_\tau$ required to decrease the pairing duration threshold, $\tau_n$, is held fixed. These results show that in the AIS the impact of $T_S$ in results in extremely reduced. This is due to the fact that education in the AIS has memory making it less critical to present a large number of samples in a short time interval. Therefore, for the new strategy, the impact of $T_S$ variations is small. The values of $T_S$ used in this plots is: $T_S = 20, 50, 100, 400, 800, 2000, 5000$. In each point,an average over 10 different tests involving different normal samples, have been calculated.

## SM6-Impact on results of varying $W_d$

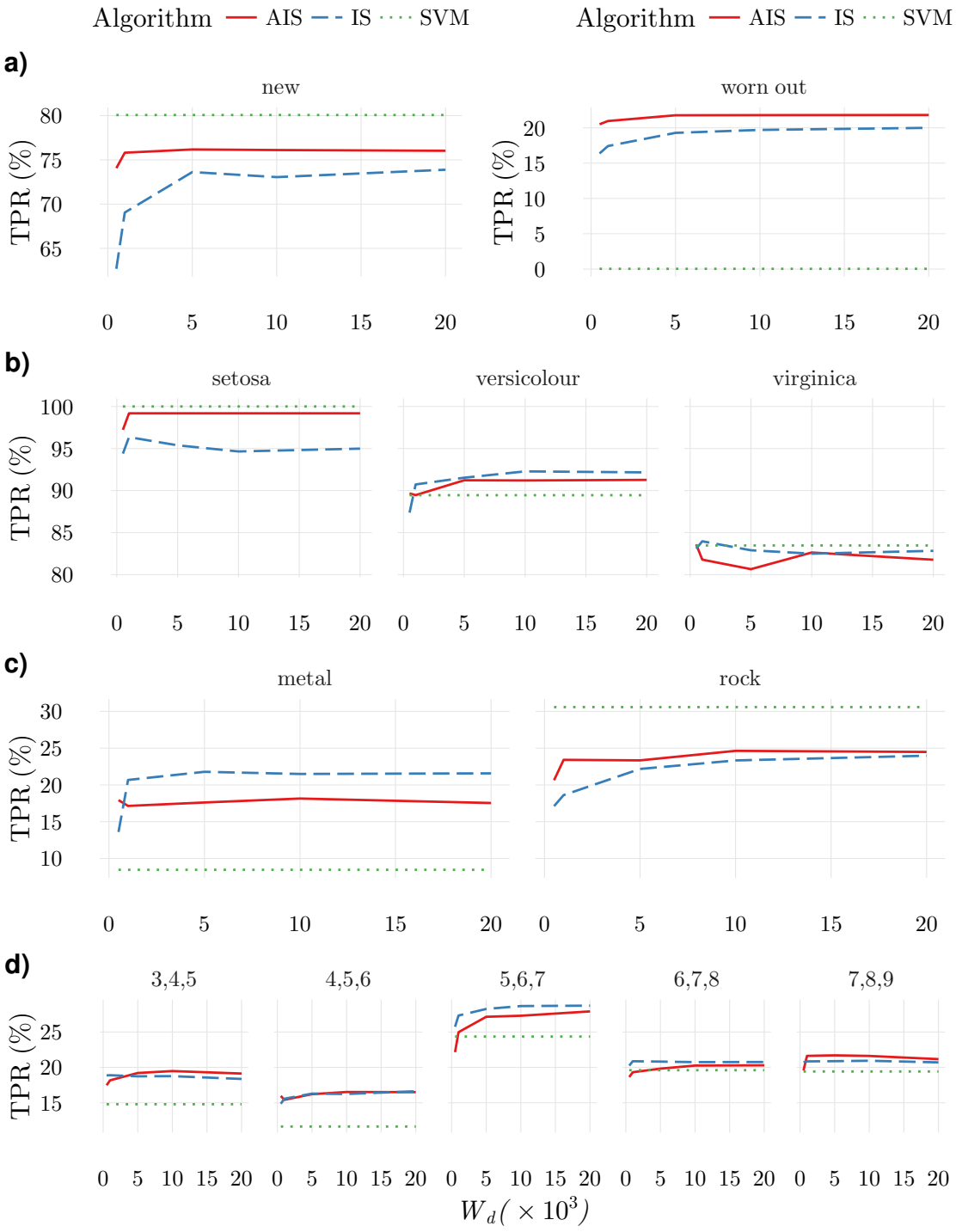


Figure 6.14: Average $TPRs$ obtained for different values of $W_d$. Overall these results show that beyond $W_d \sim 5000$ iterations, the performance of the method becomes stable. Therefore, $W_d$ is also a parameter requiring no tuning. In the results reported in this paper we used $W_d = 10000$.

## SM7-Impact on results of varying $\tau_A$

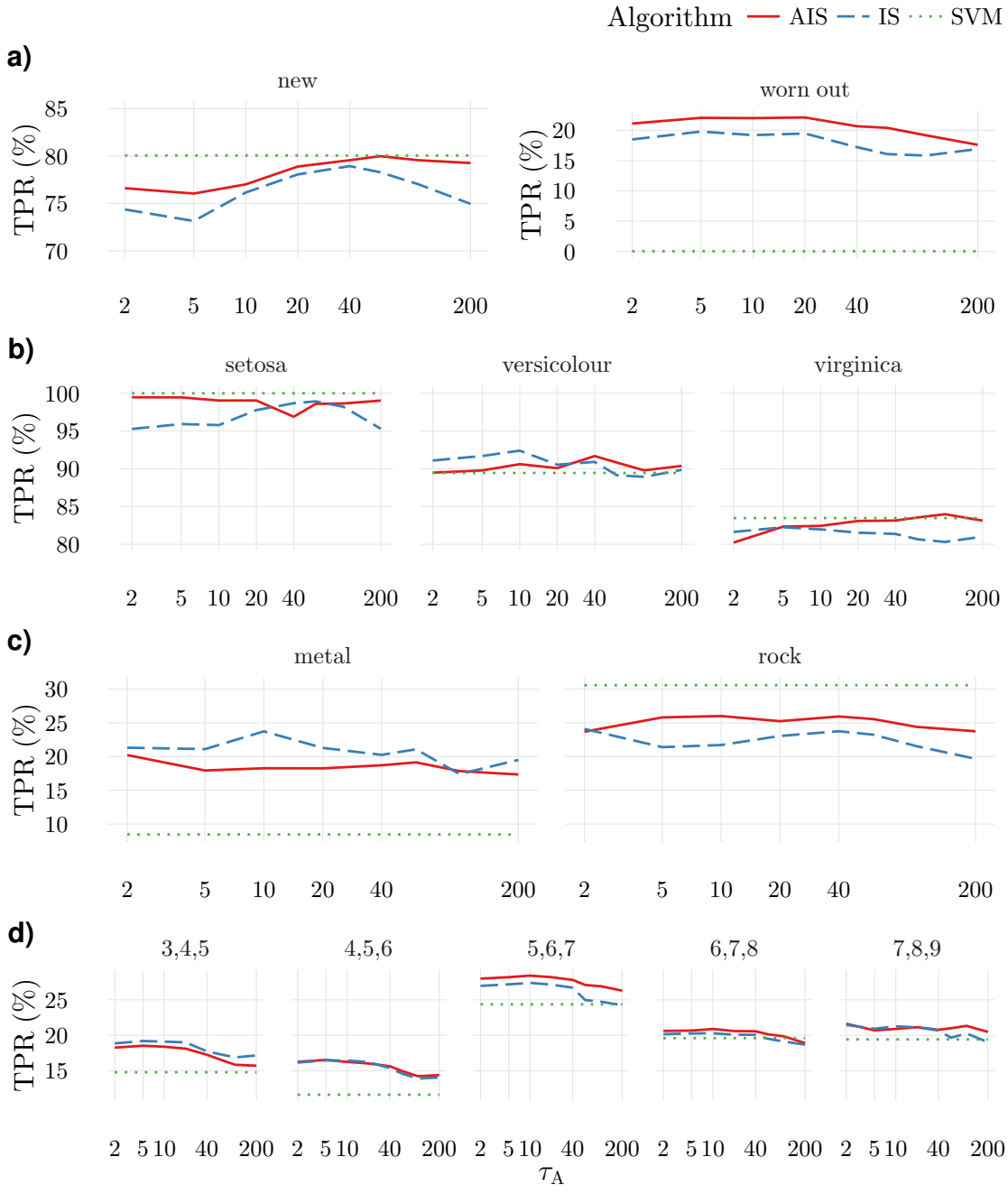


Figure 6.15: Average true positive rates obtained for different values of $\tau_A$ (and consequently $\tau_{act}$, since $\tau_{act} = \tau_A$). By order of appearance, a), b), c), d), describe the results for the ball bearings, iris, sonar and wine quality datasets, respectively. The title of each plot represents the dataset class used as the normal class, while the remaining were considered as abnormal. All results were obtained considering a 10% false positive rate. Overall discrimination improves for $\tau_A >= 5$ which implies that discrimination in these datasets is mostly due to increases of rare ligands.

## SM8-Impact on results of varying $f$

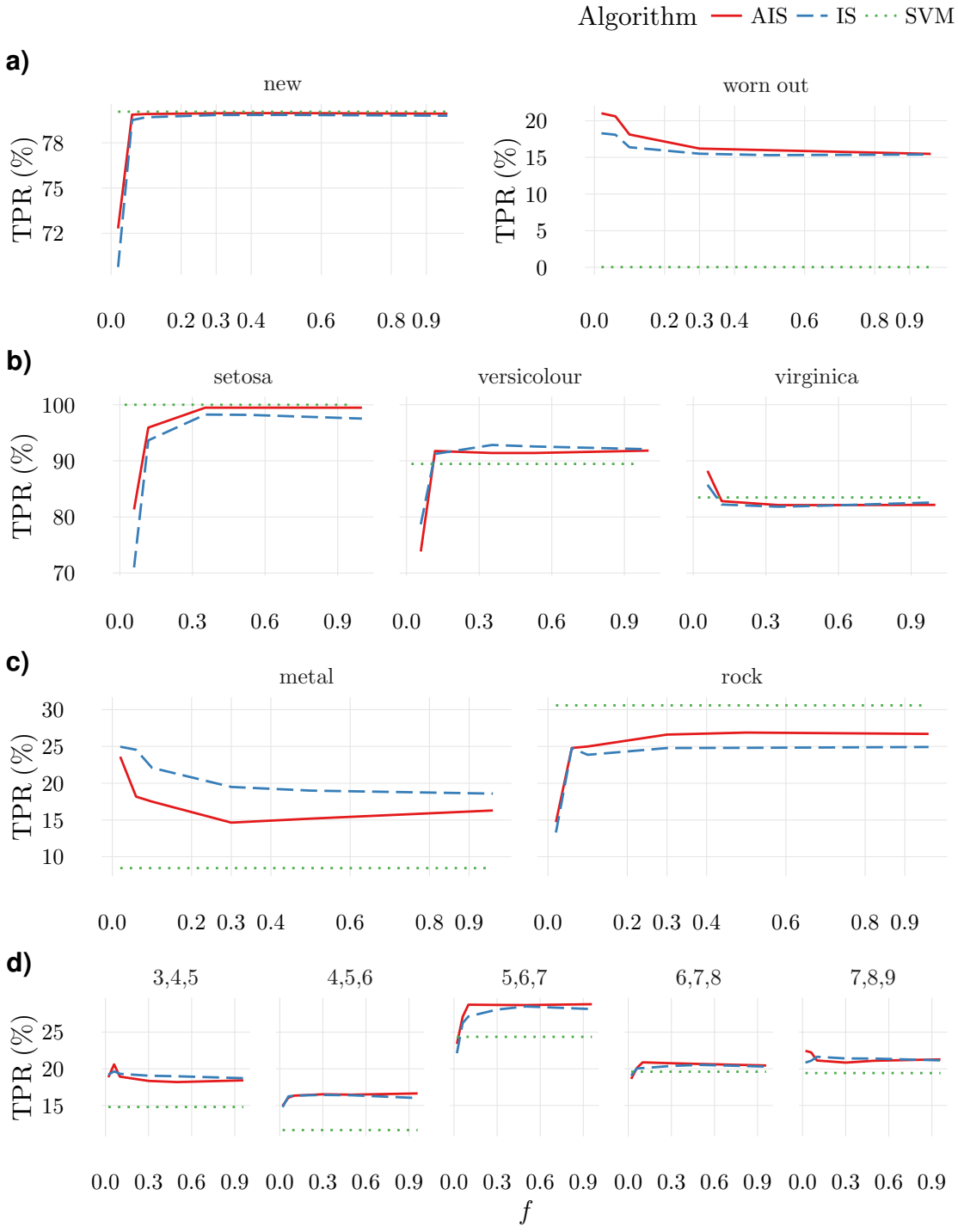


Figure 6.16: Impact of the variation of $f$ on the $TPR$ for a $10\%FPR$. Overall these results show that the maximum value of $f$ could be used. This implies that the calibration stage plays almost no role and could therefore be eliminated. This is an interesting improvement in terms of the computational efficiency of the algorithm.

## SM9-Impact on results of different SVM kernel choices

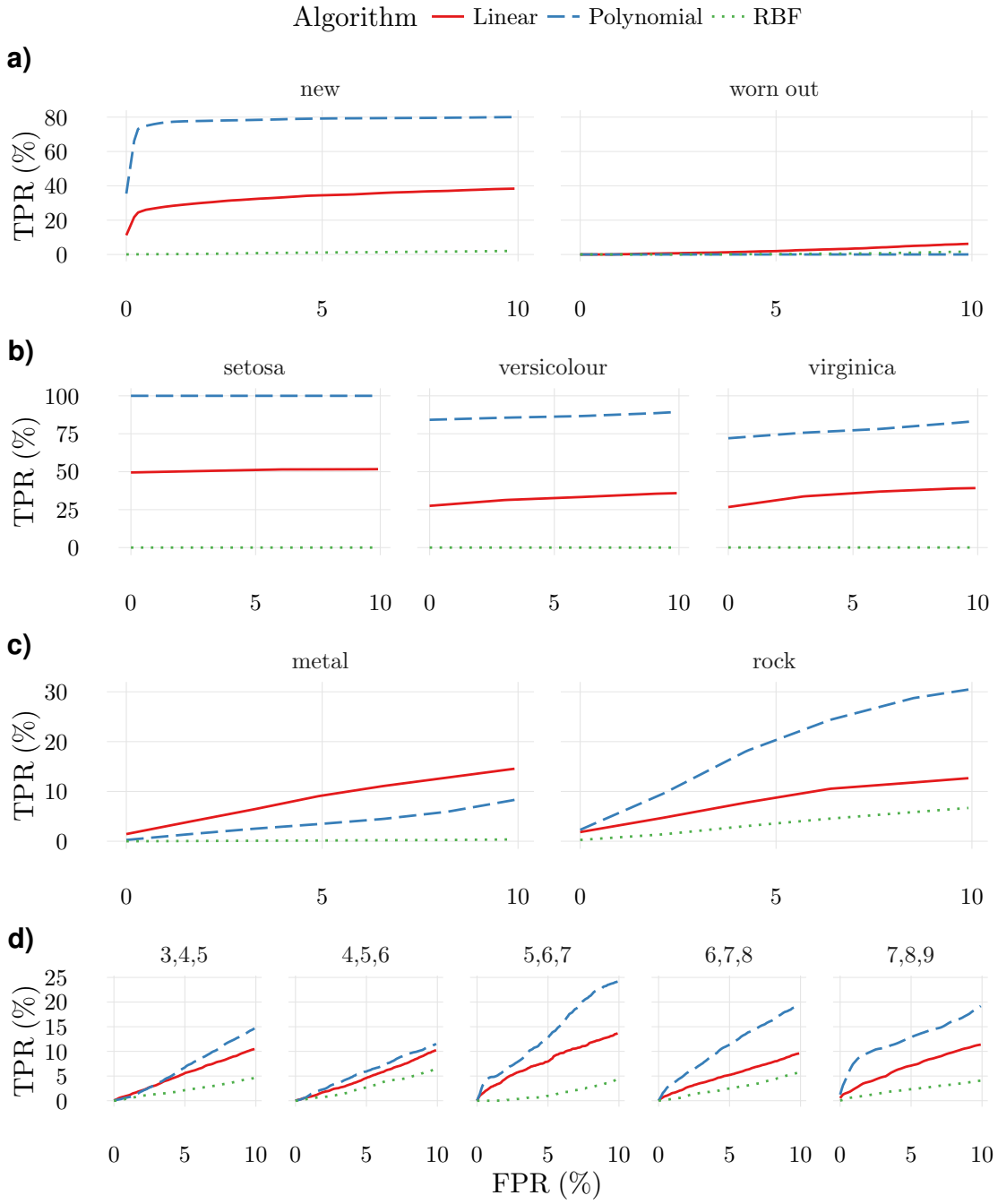


Figure 6.17: Impact of the kernel choice on the average TPR for the different datasets considered. Overall discrimination is best when the polynomial kernel with degree 2 is used. This was the kernel used in the results reported in this article. In these results, the $\gamma$ parameter used for the Gaussian kernel was $\gamma = 1/\text{dataset dimension}$, while the polynomial and linear kernels took $c = 0$. All the results were obtained training models with $v = 1/N_f$. These are the default parameters considered in [36].

## 6.9 Bibliography

[1] B. F. Faria and F. Vístulo Abreu. Cellular frustration algorithms for anomaly detection applications. *(submitted)*, 2016.

[2] F. Vístulo de Abreu, E. N. M. Nolte-'Hoen, C. R. Almeida, and D. M. Davis. Cellular Frustration: A New Conceptual Framework for Understanding Cell-mediated Immune Responses. In *Proceedings of the 5th International Conference on Artificial Immune Systems*, ICARIS'06, pages 37–51, Berlin, Heidelberg, 2006. Springer-Verlag.

[3] F. Vístulo de Abreu and P. Mostardinha. Maximal frustration as an immunological principle. *Journal of The Royal Society Interface*, 6(32):321–334, 2009. ISSN 1742-5689.

[4] P. Mostardinha and F. Vístulo de Abreu. Positive and negative selection, self-nonself discrimination and the roles of costimulation and anergy. *Scientific Reports*, 2:769, oct 2012. ISSN 2045-2322.

[5] Wangpeng He, Yanyang Zi, Binqiang Chen, Feng Wu, and Zhengjia He. Automatic fault feature extraction of mechanical anomaly on induction motor bearing using ensemble super-wavelet transform. *Mechanical Systems and Signal Processing*, 54-55: 457–480, 2015. ISSN 0888-3270.

[6] Jun Zhao, Kai Liu, Wei Wang, and Ying Liu. Adaptive fuzzy clustering based anomaly data detection in energy system of steel industry. *Information Sciences*, 259: 335–345, 2014. ISSN 0020-0255.

[7] O. P. Popoola and K. Wang. Video-Based Abnormal Human Behavior Recognition 2014; A Review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):865–878, Nov 2012. ISSN 1094-6977.

[8] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly Detection and Localization in Crowded Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32, Jan 2014. ISSN 0162-8828.

[9] Julián Candia, Marta C González, Pu Wang, Timothy Schoenharl, Greg Madey, and Albert-László Barabási. Uncovering individual and collective human dynamics from mobile phone records. *Journal of Physics A: Mathematical and Theoretical*, 41(22): 224015, 2008.

[10] Y. J. Lee, Y. R. Yeh, and Y. C. F. Wang. Anomaly Detection via Online Oversampling Principal Component Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 25(7):1460–1470, July 2013. ISSN 1041-4347.

[11] G. Creech and J. Hu. A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguous and Discontiguous System Call Patterns. *IEEE Transactions on Computers*, 63(4):807–819, April 2014. ISSN 0018-9340.

[12] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle. GrIDS – A Graph Based Intrusion Detection System for Large Networks. In *IN PROCEEDINGS OF THE 19TH NATIONAL INFORMATION SYSTEMS SECURITY CONFERENCE*, pages 361–370, 1996.

[13] Stephen Marsland. Novelty Detection in Learning Systems. In *Neural Computation Surveys*, 2003.

[14] Michael-Paul Schallmo, Scott R. Sponheim, and Cheryl A. Olman. Abnormal Contextual Modulation of Visual Contour Detection in Patients with Schizophrenia. *PLoS ONE*, 8(6), 06 2013.

[15] Michael P. S. Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terrence S. Furey, Manuel Ares, and David Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences*, 97(1):262–267, 2000.

[16] F.Y. Edgeworth M.A. XLI. On discordant observations. *Philosophical Magazine Series 5*, 23(143):364–375, 1887.

[17] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3):15:1–15:58, July 2009. ISSN 0360-0300.

[18] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12): 3448–3470, 2007. ISSN 1389-1286.

[19] V. Vapnik and A. Lerner. Pattern Recognition using Generalized Portrait Method. *Automation and Remote Control*, 24, 1963.

[20] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995. ISSN 1573-0565.

[21] B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, pages 582–588, 2000.

[22] David M. J. Tax and Robert P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20:1191–1199, 1999.

[23] Mennatallah Amer, Markus Goldstein, and Slim Abdennadher. Enhancing One-class Support Vector Machines for Unsupervised Anomaly Detection. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, ODD '13, pages 8–15, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2335-2.

[24] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[25] J. J. Hopfield. Kinetic Proofreading: A New Mechanism for Reducing Errors in Biosynthetic Processes Requiring High Specificity. *Proceedings of the National Academy of Sciences*, 71(10):4135–4139, 1974.

[26] André M. Lindo, Bruno F. Faria, and Fernão V. de Abreu. Tunable kinetic proofreading in a model with molecular frustration. *Theory in Biosciences*, 131(2):77–84, 2012. ISSN 1611-7530.

[27] T W McKeithan. Kinetic proofreading in T-cell receptor signal transduction. *Proceedings of the National Academy of Sciences*, 92(11):5042–5046, 1995.

[28] Shoshana D. Katzman, William E. O'Gorman, Alejandro V. Villarino, Eugenio Gallo, Rachel S. Friedman, Matthew F. Krummel, Garry P. Nolan, and Abul K. Abbas. Duration of antigen receptor signaling determines T-cell tolerance or activation. *Proceedings of the National Academy of Sciences*, 107(42):18085–18090, 2010.

[29] C.R. Almeida and F.V. de Abreu. Dynamical instabilities lead to sympatric speciation. *Evolutionary Ecology Research*, 5(5):739–757, 2003.

[30] Bruno Filipe Faria, Patrícia Mostardinha, and Fernão Vístulo de Abreu. Can the Immune System Perform a t-Test? *PLOS ONE*, 12(1), jan 2017. doi: 10.1371/journal.pone.0169464.

[31] M. Lichman. UCI Machine Learning Repository, 2013. URL `http://archive.ics.uci.edu/ml`.

[32] Ball bearings dataset from `http://fromwww.sidanet.org`. `http://homepage.tudelft.nl/n9d04/occ/index.html`. Accessed: 2016-10-10.

[33] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009. ISSN 0167-9236. Smart Business Networks: Concepts and Empirical Evidence.

[34] Ronald Aylmer Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugenics*, 7:179–188, 1936. doi: 10.1111/j.1469-1809.1936.tb02137.x.

[35] R. Paul Gorman and Terrence J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75, 1988.

[36] Patrícia Mostardinha, Bruno Filipe Faria, André Zúquete, and Fernão Vístulo Abreu. A Negative Selection Approach to Intrusion Detection. In *The 11th International Conference on Artificial Immune Systems (ICARIS 2012)*, volume LNCS 7597, Taormina, Italy, August 2012.

[37] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, May 2011. ISSN 2157-6904.

CHAPTER 7

# Intrusion detection using the cellular frustrated framework[1]

In this paper we describe the cellular frustration algorithm in the context of intrusion detection and benchmark its anomaly detection performance. The performance assessment is done by comparing the detection effectiveness of the cellular frustration algorithm with one-class support vector machines over the 1999 DARPA BSM dataset collected at MIT's Lincoln Labs. The results indicate that the cellular frustration algorithm can be successfully applied in the context of intrusion detection, and in some instances exhibit an anomaly detection performance that is 1.5 times the performance achieved by one-class support vector machines.

## 7.1 Introduction

The widespread use of computer systems and its increasing access simplicity creates opportunities for malicious individuals seeking to exploit the systems' security flaws and attack them for some benefit. Intrusion detection systems were created to stop the initiative of such individuals. Two general approaches can be used for computer intrusion detection: misuse detection and anomaly detection. Misuse detection triggers an alarm if a known attack signature is matched. Anomaly detection, on the other hand, generates an alarm if a given activity deviates from the normal system behaviour and, as a result, have the possibility to detect novel attacks [2].

Several data mining approaches (e.g. neural networks [3, 4], support vector machines [3, 5], random forests [6, 7], etc.) capturing different aspects of a computer system behaviour (e.g. user [8, 9], network [3, 10] or program behaviour [5, 11]) have been proposed for anomaly detection. In these approaches a model is assumed to classify new behaviour as

[1] chapter submitted as: B. F. Faria, A. Zúquete, and A. M. Lindo. Intrusion detection using the cellular frustrated framework. *(submitted)*, 2016

either normal or abnormal. However, to build the model, most of these approaches assume that training samples from both categories are available (e.g. [7]). More importantly, they assume that training samples are trustworthy, that is, the training samples' labels are considered to be correct. In practice this may not be the case and training samples may be mislabelled due to the unclear boundary between normal and abnormal categories. An obvious solution is to consider data mining approaches that can be trained with training samples from a single category, i.e. the normal category.

In this paper, we address the problem of anomaly detection with algorithms that use only the normal category for training. However, rather than focusing on the traditional data mining methods, we will use the cellular frustration algorithm, which is an immunology-inspired algorithm. Just like an intrusion detection system, the task of an immunological system is to discern normal from abnormal. Such distinction is performed with several properties that we believe are desired, such as distributed and probabilistic detection and the system is designed to recognize virtually any foreign sample, and not just those seen during training. Here, we show how the cellular frustration algorithm can be used to perform anomaly detection. We evaluated its anomaly detection performance with the 1999 DARPA BSM data, and we benchmark it with support vector machines (SVMs). Our experiments show that the cellular frustration algorithm can be successfully applied to computer security exhibiting in the best case scenario 1.5 times the anomaly detection performance of support vector machines.

The rest of this paper is organized as follows. In section 7.2 we review some of related work. In section 7.3 we describe the cellular frustration algorithm and in section 7.4 we describe the methods used for comparison (support vector machines). Section 7.5 describes the approaches and information used for creating the profiles which are then used by the algorithm. Finally, in section 7.6 we present the results and in section 7.7 we discuss our contribution.

## 7.2 Related work

The DARPA datasets constitute one of the first attempts to provide standard datasets for benchmarking intrusion detection systems. This has led to a wide adoption by the research community and resulted in the implementation of various data mining approaches modelling either programs' behaviour [5, 11–14] or network behaviour [3, 4, 6, 7, 15, 16]. Data mining approaches, such as random forests [6, 7], support vector machines [3, 5], neural networks [3, 4] and fuzzy logic [15, 16] have been implemented and compared. The biggest difference between all these methods relies on what they are modelling and how. In this paper we focus on program behaviour, and as a result we will only describe the different approaches to model program behaviour.

There are essentially two different approaches for modelling program behaviour: Forrest *et al.* [14] and Liao's *et al.* [13]. Both Forrest *et al.* and Liao *et al.* rely on system calls' information. The Forrest *et al.* approach considers that the sequence of system calls made

by a program during an intrusion is different from those made during the program's normal operation. Liao *et al.*, on the other hand, drew an analogy between a text document and the sequence of system calls made by a program. They considered that the frequency of system calls during the execution of a program during its normal functioning differs from those during an intrusion. Independently of the approach, each program will have a set of profiles that characterize it.

This paper considers both approaches and compares the intrusion detection performance of the cellular frustrated algorithm and support vector machines over the training data of program behaviour profiles.

## 7.3 The Cellular Frustration Algorithm

The cellular frustration algorithm was introduced by Abreu *et al.* [17] as an attempt to explain how the human adaptive immunological system works. The algorithm comprises a set of agents in interaction to explain the mechanisms of tolerance, towards the body own healthy cells, and reaction against foreign pathogenic cells. In Abreu's *et al.* point of view, the problem tolerance and reaction depends on the time that agents spent paired with each other. For instance, if two agents are paired more than a threshold amount of time then, an immune response is developed which would not happen otherwise. However, this raises the question on how to achieve different pairing times for the body's own healthy cells and the pathogenic cells? This question is addressed in the following paragraphs, where we briefly explain the cellular frustration algorithm.

The cellular frustration algorithm is rooted in the stable marriage problem (SMP) whose solution, due to Gale and Shapley, awarded Shapley the 2012 economics Nobel prize. In the SMP there is a set of $n$ men and a set of $n$ women where each man/woman has an ordering preference for the elements of the opposed set. The aim is to marry all men and women such that no man and woman in two distinct marriages prefer to be married with one another than with the current ones.

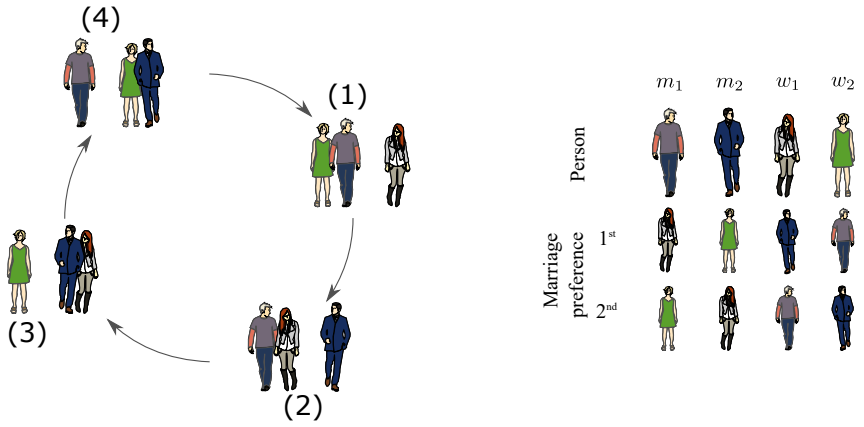


Figure 7.1: Illustration of the agents' dynamics

In the cellular frustration algorithm, women and men purport the roles of presenter and detector cells, i.e. APC (Antigen-Presenting Cell) and T cells, respectively, but with a different aim. Instead of being concerned with finding stable matches, the cellular frustration algorithm is interested in the marriage-divorce dynamics. To easily convey this concept to the reader a simple illustration is used. Consider that there is a set of 2 men and a set of 2 women with ordering preferences, as defined in Figure 7.1. Here, man 1 ($m_1$) prefers woman 1 ($w_1$) to woman 2 ($w_2$), man 2 ($m_2$) prefers woman 2 ($w_2$) to woman 1 ($w_1$), and the same reasoning goes for the women.

Starting with marriage (1) illustrated in Figure 7.1 by the marriage between man 1 $m_1$ and woman 2 $w_2$. In this marriage, if woman 1 $w_1$ proposes to $m_1$, then $m_1$ will face a decision – either he continues married to $w_2$ or divorces $w_2$ and marries $w_1$. However, since $w_1$ ranks higher in $m_1$'s preference list, then $m_1$ will break the marriage with $w_2$ and marry $w_1$ forming marriage (2). Applying the same reasoning to the remaining marriages, one can see that all marriages will be short-lived.

Consider now the situation where an impostor woman tries to pose as woman 1. Consider also that this impostor knows how to exhibit the same traits as women 1 (i.e., from the men's point of view it is woman 1) but does not know woman 1 preferences for men. In this simple example two situations arise: either the impostor chooses the same ordering for men as woman 1 or chooses the opposite ordering. If the impostor chooses the same ordering as woman 1 (an unlikely scenario when the number of men and women are much bigger), nothing is changed from the previous case. However, if the opposite ordering is chosen, then the cycle in Figure 7.1 is broken, as the step from marriage (2) to marriage (3) ceases to exist - man 1 is ranked higher in the intruder preference list. As a result, marriage (2) will be long-lived in comparison to any marriage when there is no intruder. This difference in the liveness time of the marriage is the basis for pathogen detection according to the cellular frustration algorithm.

The watchful reader may argue that for this hypothesis to work there must always exist unengaged men and women. In fact, if the above example would have started with a configuration where all men and women were already engaged, then no couple changes would take place. This is the situation of stable matching [18]. However, to guarantee that this situation does not happen certain measures need to be taken, such as the introduction of a dissociation probability [19].

## Cellular frustration framework as a data mining tool

So far our concern was to explain the fundamental concepts underneath the cellular frustration algorithm. In this subsection we show how those concepts have been materialized in a data mining framework [20, 21].

We start by reinforcing the idea that both men and women are agents that interact with each-other and not simply data. The interactions between both agent types depend on the information exhibited. As a result, the first thing to be explained is the information displayed by each agent type (Figure 7.2). In this algorithm, men act as the detectors

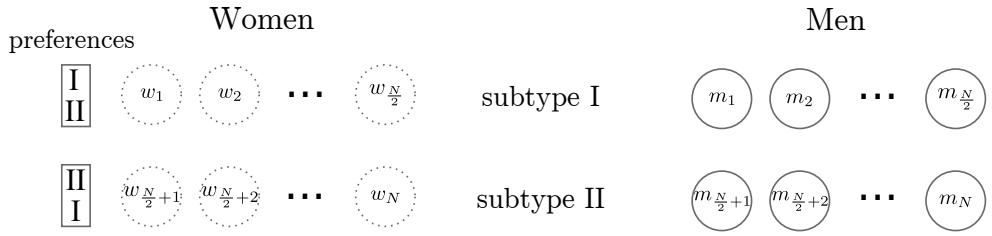


Figure 7.2: Agents arrangement and information. Women of subtype I prefer men of subtype I to subtype II and vice versa for women of subtype II. Men classification of women depends on the information exhibited by women.

and are grouped in two evenly distributed subtypes – subtype I and subtype II. Women preferences for men are expressed in terms of these subtypes and are fixed throughout the execution of the algorithm. The number of agents from each subtype, that is men and women, is always even. As a consequence, there is half women preferring men of subtype I to subtype II and half women preferring subtype II men to subtype I. Women preferring subtype I men are said to belong to subtype I, and so forth.

Women, on the other hand, act as presenters, that is they present the information of a sample. As a result, women are ranked in men's preference lists based on the information they present. Since, that each woman presents a different sample feature, men's preference lists have to rank every possible woman presenting value, which is infeasible. The approach considered in [20] to solve this problem was to express men's preference lists in terms of women subtypes. However, instead of grouping all women in two subtypes, women subtypes are a result of men classification. Each man classifies each woman by the information presented in two subtypes such that no two different women present the same subtype. As a result, the number of subtypes, from the point of view of each man, is two times the number of women.

Women subtype determination is the first step in the algorithm. To easily convey how this is achieved, lets consider a set of training samples $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_d \in \mathbb{R}^n$ with respective labels $y_1$, $y_2$, ..., $y_n \in \{normal, abnormal\}$, where $d \in \mathbb{N}$ denotes the number of observations and $n$ the number of features. Training samples are presented by women to men, where each woman presents a feature of a training sample. The first step is to draw, at chance, a random value $\xi_j$ between 0 and $v$ or between $1 - v$ and 1, $0 \leq v \leq 1$ for each man $j$, $j \in [1, n]$. Then, for each feature $i \in [1, n]$ the cumulative frequency distribution of the training patterns $x^i$ is computed and the training sample value for which the cumulative frequency matches $\xi_j$ is registered. Here, the training samples with a cumulative frequency below $\xi_j$ are considered of subtype $\mathrm{I}^i$ while the training examples above are considered of subtype $\mathrm{II}^i$, which translates to women $i$ either being of subtype $\mathrm{I}^i$ or $\mathrm{II}^i$ (Figure 7.3).

The following step is training. In training each training pattern is presented by women to men and the matching dynamics is executed. Matching dynamics gives, in every iteration, an opportunity to each man and woman to interact with a random agent of the

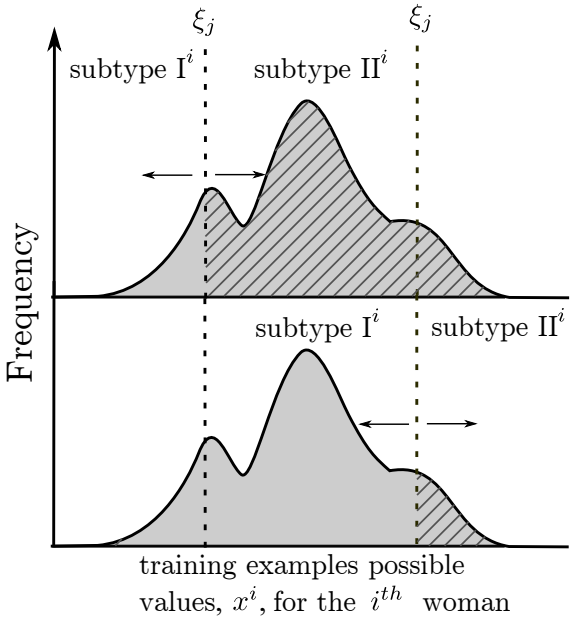


Figure 7.3: Illustration of the process to obtain women discrimination values for men.

opposite type. A new pair is formed whenever the two interacting agents prioritize the new interaction over the former ones. In the event that they were already conjugated, the former pairs are terminated and the conjugation duration registered. The iteration is incremented after all agents had an opportunity of interacting with an agent of opposite genre. From $T_S$ to $T_S$ iterations, women change the exhibited training pattern. Men whose matching durations exceed a threshold value $\tau_{NS}$ see their list replaced by another randomly drawn. Finally, the threshold value $\tau_{NS}$ is updated, if no list has been replaced, every $W$ iterations, with $W >> T_S$, and takes the value of the largest matching lifetime during the last $W$ iterations.

In [21] it was shown that training could be improved, that is require less iterations to reach a small $\tau_{NS}$. Instead of considering that all men interact with all women, it was suggested that men interact with only a fraction of the women. By not interacting will all women, men's preference lists would only have to rank a subset of the information displayed by women. As a result, the number of replacements to partially order a man preference list, and hence reach a small $\tau_{NS}$, would be smaller. This mechanism was denoted as positive selection and operates by changing the set of women a man can interact with if the time spent alone by the man does not decrease. More formally, all men who are not matched for more than $\tau_{PS}$ see the set of women they can interact with changed. As before, $\tau_{PS}$ is updated to the maximum time a man spends unmatched every $W$ iterations if the time men spend alone decreases during those $W$ iterations. Nonetheless, in this paper we assumed that each man started with a random subset of 20 women he can interact with at the start of training and no update to this subset was perform during training.

Training aims to reduce the longest matching durations that men make with women presenting *normal* information. However, it was verified that this mechanism alone was not sufficient and some men, after training, could still exhibit long matching durations with women presenting *normal* information [21]. As a result, a new mechanism designated by *anergy* was introduced in the algorithm. This mechanism considers the independent training of several sets of men with the same connectivity – men don't interact with all women, just a fraction [21] – denoted as detector repertoire. Then, when a new sample is to be tested, a set is randomly chosen to start the matching dynamics, during which all matchings exhibiting a long matching duration ($> \tau_A$) are broken apart and the men replaced by equivalent men from a randomly chosen set of the detector repertoire. The idea of this mechanism is simple, if a man from a given set evades training the probability that it does so in another set is small. As a result, men will only exhibit long matching durations towards women presenting *abnormal* information.

After training, the algorithm is ready for testing new samples. Testing new samples is just a matter of executing the matching dynamics (with the mechanism of *anergy* explained above) for each new pattern. Then, for each man, $i$, the frequency $F^i_{>\tau}$ for making a matching duration $M^i_j$, $1 \leq j \leq W$, bigger than $\tau$ is compared with the threshold frequency for making the same matching duration at the calibration stage. Mathematically, this frequency is expressed as:

$$F^i_{>\tau} = \frac{\sum_{k=\tau}^{W} M^i_k}{\sum_{k=1}^{W} M^i_k}. \tag{7.1}$$

If the sum of all frequencies for the new pattern is bigger than the sum of threshold frequencies at the calibration stage then this pattern is classified as an anomaly, otherwise it is classified as normal.

Calibration is the last step of training and amounts to execute the matching dynamics with *anergy* for each training sample and registering the sample frequency of each man to make a matching duration of at least $\tau$ iterations. Men threshold frequency is then determined by ordering the men frequencies over all samples and choosing the 10% highest value for each man. This value corresponds to consider the 10% highest matching duration and it was chosen to agree with the result used on [20]. Also, to maintain coherency with [20] $\tau = \tau_A$ was set to 5 iterations.

Here, we have given just a brief explanation of how the cellular algorithm works. For a thorough description of the algorithm we direct the reader to [21] and [20].

## 7.4 Support Vector Machines

In the previous section we gave a brief description of the cellular frustration detection algorithm. The aim of this section is to describe the algorithm used for comparison. The choice fell over support vector machines (SVMs) since they have already been successfully applied to the 1998 DARPA data set from MIT Lincoln Laboratory [5].

Support vector machines were introduced by Vapnik *et al.* [22] as a method to perform binary classification. The underlining idea consists in finding a separating hyperplane that maximizes the separating margin between both categories. The feature vectors that lie on the defined maximum separating margin are denoted as support vectors. Hence, classifiers that exploit this property are designated as support vector machines.

Since the initial formulation by Vapnik several improvements and extensions to the model have been performed, namely: (i) soft margin SVMs for the non-separable case (i.e., with mislabelled data); (ii) nonlinear SVMs for nonlinear separable data; and (iii) one-class SVMs for the case where the hyperplane has to be defined from a single category. The last extension is particularly relevant because it allows a fair comparison between the performances of both detection algorithms (i.e., a comparison independent of the information used).

## One-class SVM

There are essentially two different approaches for defining a one-class SVM: the one of Schölkopf *et al.* [23] and the one of Tax and Duin [24]. Both approaches consider that anomalies are not concentrated [25–27] and differ only on the method to obtain the closed class boundary of concentrated normal data. Schölkopf *et al.* consider the origin as the only member of the anomaly class, $-1$, and the aim is to separate the normal class data from the origin with maximum margin. Tax and Duin, on the other hand, assume a spherical boundary encompassing the normal class data $+1$, and aim to minimize the hypersphere volume so that the probability of including outliers is minimized [24].

In this article, the Schölkopf *et al.* approach was used because it is the default method used in the Libsvm implementation [28]. Consequently, we dedicate the next paragraphs to briefly describe their approach.

Consider a set of training samples $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, ..., $(\mathbf{x}_d, y_d) \in \mathbb{R}^n$, where $d \in \mathbb{N}$ denotes the number of observations. First the training samples are mapped into a high dimensional feature space $F$ using a function $\boldsymbol{\phi}(\cdot)$. This function transforms the otherwise nonlinear, separable data into linear separable data by means of a simple kernel evaluation [29]:

$$k(\mathbf{x}, \mathbf{y}) = (\boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{y})) \tag{7.2}$$

such as the Gaussian kernel:

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{c}} \tag{7.3}$$

Then, in $F$, the hyperplane that separates the training examples from the origin with maximum margin needs to be determined. Hyperplane determination can be accomplished by solving the following quadratic optimization problem:

$$\begin{aligned} \min_{\mathbf{w}\in F,\xi\in\mathbb{R}^d,\rho\in\mathbb{R}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{vd}\sum_i \xi_i - \rho \\ \text{subject to} \quad & (\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}_i)) \geq \rho - \xi_i, \xi_i \geq 0. \end{aligned} \tag{7.4}$$

Here, nonzero slack variables $\xi_i$ penalize outliers in the objective function. That is, if $\mathbf{w}$ and $\rho$ solve the optimization problem, then $f(\mathbf{x}) = \text{sign}(\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}) - \rho)$ will be positive for most training examples $\mathbf{x}_i$, and at the same time, the support vector regularization term $\|\mathbf{w}\|$ will be small. The trade-off between these two goals is regulated by the parameter $v \in (0, 1)$.

The optimization problem (7.4) can, in principle, be solved directly with numerical methods. However, this approach is impractical for a large or even infinite dimensional $\boldsymbol{\phi}(\cdot)$ space. The solution, found by Boser *et al.* [29], consists in transforming (7.4) into its dual by means of the Lagrangian and expressing it in terms of a kernel (7.2), resulting in:

$$\begin{aligned} \max_{\boldsymbol{\alpha}\in\mathbb{R}^d} \quad & -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) \\ \text{subject to} \quad & \sum_i \alpha_i = 1, 0 \leq \alpha_i \leq \frac{1}{vd}, \end{aligned} \tag{7.5}$$

which, when solved yields the decision function:

$$f(\mathbf{x}) = \text{sign}(\sum_i \alpha_i k(\mathbf{x}_i,\mathbf{x}) - \rho). \tag{7.6}$$

The decision function (7.6) is expressed in terms of the Lagrange multipliers $\alpha_i$, the training patters $x_i$ and $\rho$. The $\rho$ parameter can be obtained by exploiting the fact that, at the optimum, any $\alpha_i$ that satisfies $0 < \alpha_i < \frac{1}{vd}$ has a corresponding training pattern $\mathbf{x}_i$ that is a support vector and satisfies:

$$\rho = (\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}_i)) = \sum_j \alpha_j k(\mathbf{x}_j,\mathbf{x}_i) \tag{7.7}$$

Finally, the classification of new patterns $\mathbf{x}$ is just a matter of verifying the sign of the decision function.

## 7.5 Dataset analysis

The 1999 Darpa dataset represents one of the first attempts to provide a standard dataset for the evaluation of intrusion detection systems [30]. This dataset was built by collecting information from several sources during a 5 week period on a simulation network that simulated the traffic of an Air Force local area network. The sources of information are: tcpdump data from inside and outside sniffers, Solaris Basic Security Module (BSM) audit data collected from a Solaris host and Windows NT audit event logs collected from a Windows NT host (Figure 7.4). Weeks 1 to 3 are considered as training data, and week 4 and 5 as testing data. Week 1 to 3 can be further divided in attack free (week 1 and 3)

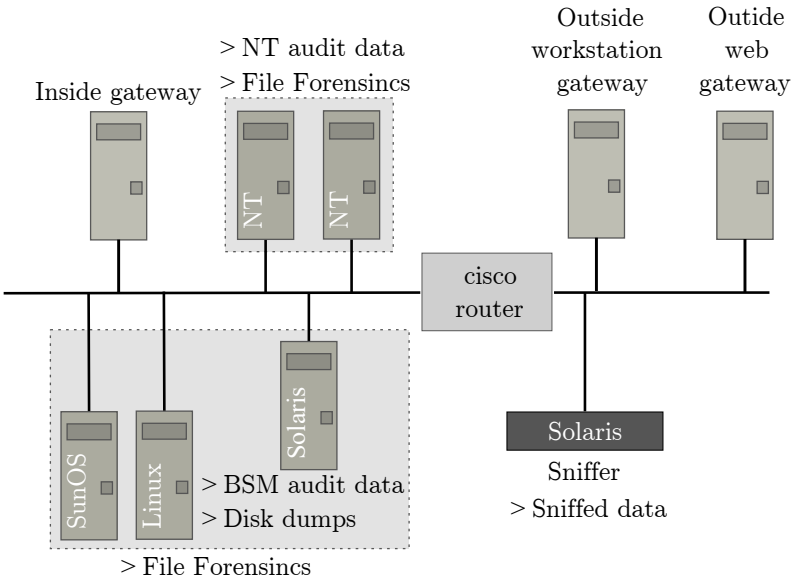


Figure 7.4: Illustration of the physical network used to simulate the traffic of an Air Force local area network. The network included inside and outside sub-networks separated by a router. The outside sub-network included two workstations simulating gateways to a virtual, outside Internet provider. The inside sub-network includes victim machines of different types (e.g. Linux, Solaris, Sun OS) and a gateway to various other inside workstations. The data used in our work was collected from the inside victim running Solaris.

and not attack free (week 2). In weeks 4 and 5 a number of attacks, described in [31], have been conducted.

In this paper we used the BSM audit data from the Solaris victim. The BSM audit data contains information about the system calls made by programs running on that machine. The most pertinent information for our detector is: (i) the system call (event); (ii) the process number that made the system call (pid), (iii) the session number which registers the TCP/IP connection made to the computer (sid); and (iv) the system call error code.

The two approaches considered for transforming the BSM audit data into a set of samples were already described in the literature and are: system calls' sequence [14] denoted by us as Forrest method and system calls' frequency [13] denoted as L&V method. However, for completeness we also outline them here (following subsections).

After generating the set of samples using the above methods, each sample was then labelled as *normal* or *abnormal*. Sample labelling was accomplished by matching the unix time-stamp of the system calls involved in sample generation with the time-stamp in the identification scoring truth. If the time-stamp of any of the system calls used to build the sample matches any of the time-stamps in the identification scoring truth the sample is marked as *abnormal*, otherwise, it is marked as *normal*.

## Forrest method

In [14] the authors advocate using a sequence of $n$ system calls to build a program's profile. The program's profile set is built by $k$-step, $n$-wide sliding window ($k < n$) over the complete program's sequence of system calls (see Figure 7.5). This step was considered for every program (PID) in every session (SID). However, we went one step further, since we also considered the system calls error code as a possible additional source of information.

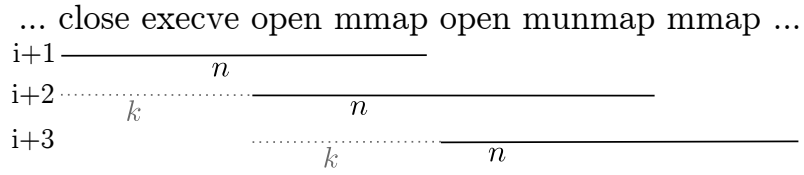


Figure 7.5: Illustration of profile generation from a program's sequence of system calls. A new profile $i+1$ is build considering a jump of $k$ system calls from the start of the previous profile $i$ and gathers information of $n$ system calls.

Several tests have been performed with this approach. The aim was to choose the parameters $k$ and $n$ that in conjunction with the system call information give the best detection ratios. The tests considered profiles with (i) the system call number only and (ii) the system call number and error code. We started by evaluating the influence of $k$ and the system call information on detection performance. As a result for each type of profiles – (i) and (ii) – $n$ was fixed at 80 and $k$ was varied using the following values: 4, 8, 16, 24, 32 and 64. After evaluating the impact of $k$ on the different type of profiles, the impact of $n$ was evaluated for the test that provided the best detection ratio.

## L&V method

Liao *et al.* [13] considered an approach where system calls' frequencies, instead of short sequences of system calls, are used to represent program behaviour. Two techniques were employed, namely frequency weighting (*tf*) and *tf-idf* weighting to transform the system calls of a process (PID) in a given session (SID) into a vector. Frequency weighting assigns the number of system calls during the execution of a process into a vector entry. *tf-idf* is an extension to frequency weighting that reduces the importance of system calls that are present in the corpus of programs. This technique derives from text mining, where it attempts to reduce importance of terms like "the" or "is" that can have a high frequency in a document but are not necessarily important in the corpus of documents. Finally, the length of the process vectors is equal to the number of unique system calls, which for this exercise is 56. Contrary, to the Forrest method, there are no parameters that can be varied. The additional information of the system call return error code has not been considered. This was because contrary to system calls, we believe that there is no additional information, at least in terms of frequency, on the system call return error code.

That is, under normal operating conditions, system call errors should be sporadic, thus not characteristic.

In the next section we present the results pertaining to both approaches.

## 7.6 Results

Several tests have been conducted using both methods (Forrest and L&V method) to pre-process data. The main goal of the tests is to assess if the cellular frustration algorithm is a good candidate for intrusion detection. In each test, 10 different sets containing profiles from week 1 and 3 were used for model training. Each set was constructed by selecting at random 50% of the available profiles for week 1 and 3. The remaining profiles were used in conjunction with the profiles of week 4 and 5 for testing.

Table 7.1: Cellular frustration algorithm settings

| | |
|---|---|
| $W$ | $10^5$ iterations |
| $T_S$ | 100 iterations |
| Training duration | $10^9$ iterations |
| Number of detector repertoires | 6 |
| $v$ | 20% |
| Anergy time | 5 iterations |

Model training was performed with the default parameters. In the case of support vector machines this means using the radial kernel and $v = 0.5$, while for the cellular frustration algorithm it means using the parameters specified in Table 7.1. The results are presented in the next two subsections followed by a discussion. All results are shown in terms of receiver operating characteristic curves (ROC) even though their use is not advised, in the context of intrusion detection, for single model performance evaluation [32]. However, when the task is to select classifiers based on their performance ROC curves are a more powerful tool than simple accuracy metrics [33]. Finally, we outline the fact that the following plots are all limited to a false positive in the range $[0, 10]\%$, because in the context of intrusion detection higher values represent an unmanageable event alert rate [32].

### Forrest method

The first results, comparing Forrest *et al.* sequence of system calls approach, can be appreciated in Figure 7.6. Distributed horizontally plots show the Receiver Operating Characteristic (ROC) curves for the different $k$ values, while vertically distributed plots show the ROC curves when the system call return error code is taken or not into account. In grey the standard deviation in the true positive rate that resulted from the 10 different set of profiles used for model training.

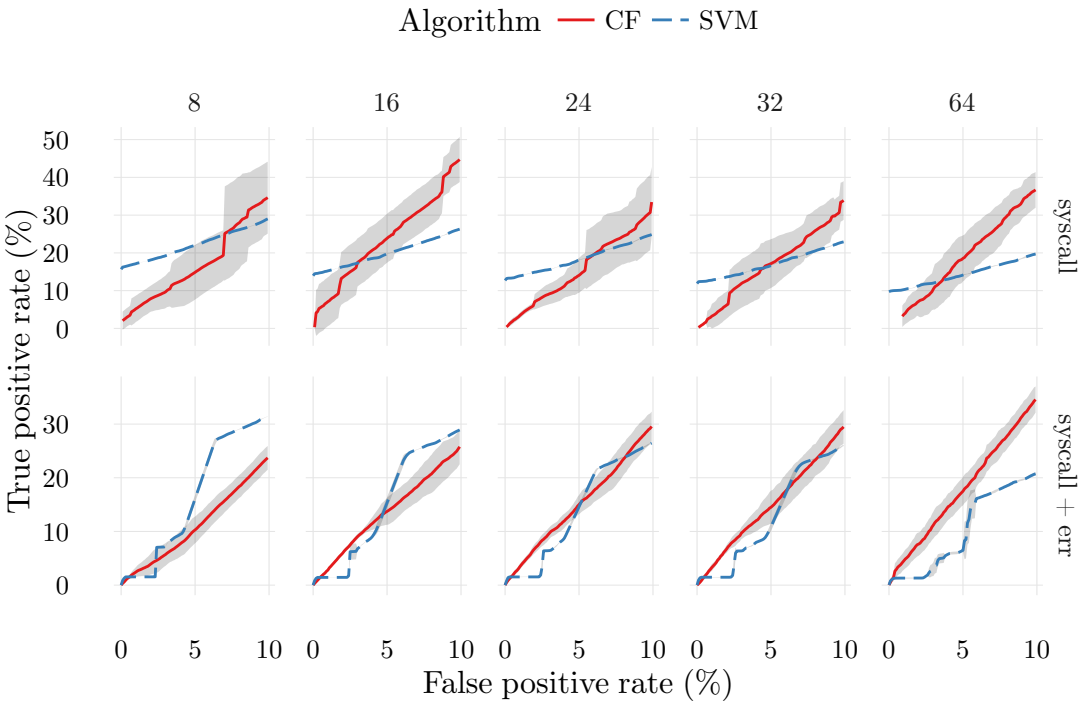


Figure 7.6: Receiver Operating Characteristic (ROC) curves comparing both algorithms for the different scenarios of $k$, $k = [8, 16, 24, 32, 64]$, and information used to build the profiles – system calls' numbers and system calls' numbers and return error code. At grey the standard deviation in the true positive rate considering the 10 different training profiles.

First, we highlight the capability of SVMs that, even when different sets are used for model training, they produce consistent results. This in turn results in a small standard deviation of the true positive rate (less than 2%). The same is not true for our CF algorithm, which produces in the worst case a standard deviation of 22% in the true positive rate.

In both scenarios, that is considering and not considering the system calls' error code, the increase of $k$ in SVMs leads to a decrease in the detection capability. By contrast, the CF algorithm registers a maximum of detection when $k$ is 16 (considering only the system call) and $k$ is 64 (system call plus return error code).

The addition of the system calls return error code results in different behaviours for both models. With SVMs it decreases the detection capability. For instance, considering only the system call code and a 0% false positive rate, SVMs have a true positive rate higher than 10%, which decreases to ∼1% when the system call return error code is considered. Even though not as severe, the same effect is also present in the CF algorithm. Nonetheless, the most perceptible effect is the decrease of the standard deviation in the true positive rate, that is, it makes the results more consistent and independent of the training set.

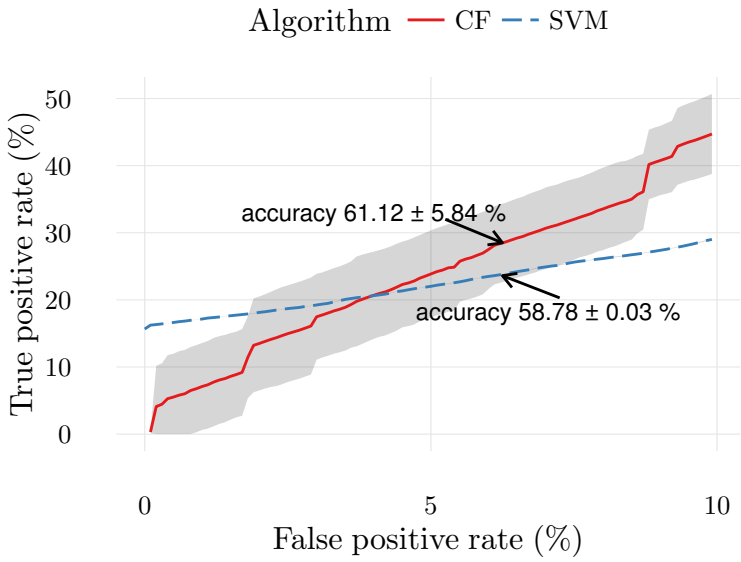


Figure 7.7: Receiver Operating Characteristic (ROC) curves comparing the best response from both algorithms. In both cases only the system calls' code was considered but with different values for $k$. For SVMs the $k$ considered was 8 while for the CF algorithm the considered $k$ was 16.

Next, we compare the best ROC curves achieved by each algorithm. That is, considering only the system calls' number and $k = 8$ for SVMs and $k = 16$ for CF algorithm. This curves are shown on Figure 7.7.

For a false positive rate below 3% SVMs perform better than the CF algorithm. In fact, for a false positive rate of 0% SVMs exhibit a true positive rate of 15.66% while the CF algorithm 0%. Only for values of false positive rates above 3% does the CF algorithm become competitive. At a false positive rate of 5% the CF algorithm exhibits a true positive rate of $23.86 \pm 6.49\%$, while the SVMs $22.02 \pm 0.4\%$. At higher false positive rates CF completely overcomes SVMs, reaching a true positive rate of $44.90 \pm 6.01\%$ for a false positive rate of 10%, while SVMs only achieve $29.14 \pm 0.05\%$.

Finally, we evaluate the impact of $n$ on the best ROC curve attained by each algorithm, that is, as above, considering only the system calls' number and $k = 8$ for SVMs and $k = 16$ for CF algorithm. Four different values of $n$ have been tested, $n = \{40, 80, 120, 160\}$, and the results shown on Figure 7.8.

The increase of size sample $n$ impacts each algorithm differently. Nonetheless, as before, the standard deviation in the true positive rate exhibited by SVMs is negligible when compared with the one exhibited by the CF algorithm. Considering the impact in the true positive rate, in SVMs the true positive rate increases with the increase of sample size, saturating at a sample size, $n$, value of 120. This contrasts with the situation of the CF algorithm where the sample size for which it attains the best trade off – high true positive rate and small standard deviation – has a size of 80.

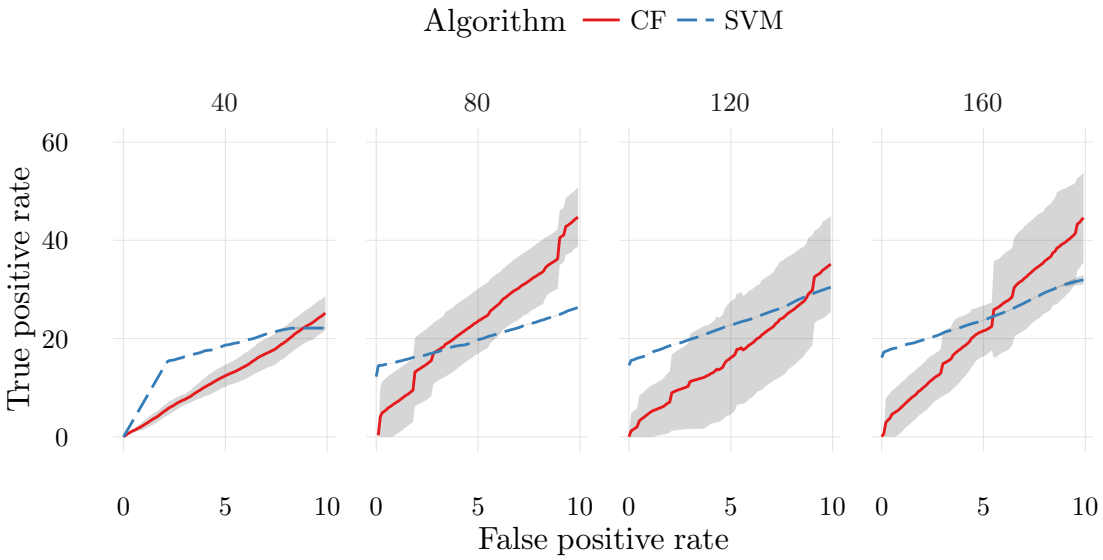


Figure 7.8: Receiver Operating Characteristic (ROC) curves comparing both algorithms for different values of $n$, $n = [40, 80, 120, 160]$, and considering only the system call's number. The value of $k$ considered was 8 for SVM and 16 for CF. At grey the standard deviation in the true positive rate considering the 10 different training profiles.

## L&V method

The results concerning Liao *et al.* approaches are shown in Figure 7.9. The plot on the left shows the ROC curves comparing SVMs and the CF algorithm when term frequency (tf) was used for profile creation. On the right we have a similar comparison but when term frequency-inverse document frequency was used. The bands mark the standard deviation on the true positive rate for each method.

We start by comparing the standard deviation in the true positive rate. Contrary to the last results, the standard deviations exhibited by SVMs for this approach are not small ($< 2\%$) and can be as high as 17.52%. In fact for both approaches, tf and tf-idf, SVMs have a higher standard deviation in the true positive rate than the CF algorithm. The maximum standard deviation exhibited by CF for the same false positive range was of 13.47%.

Algorithm performance was similar in the case of the tf approach. However, as already mentioned, SVMs exhibit a higher standard deviation in the true positive rate. Even though in this approach it was marginally higher ($< 2\%$). For instance, at a false positive rate of 0% neither CF or SVMs exhibits any detection. However, when the false positive rate is increased to 2.5% SVMs have a true positive rate of $10.47 \pm 2.37\%$ and CF of $11.10 \pm 3.62\%$. Increasing the value of false positive rate to 5% the value of standard deviation for SVMs is $18.24 \pm 7.38$ and for CF $19.71 \pm 6.71\%$. Finally, at a false positive rate 10% the values for $17.18 \pm 14.99\%$ and for CF $33.44 \pm 13.50\%$.

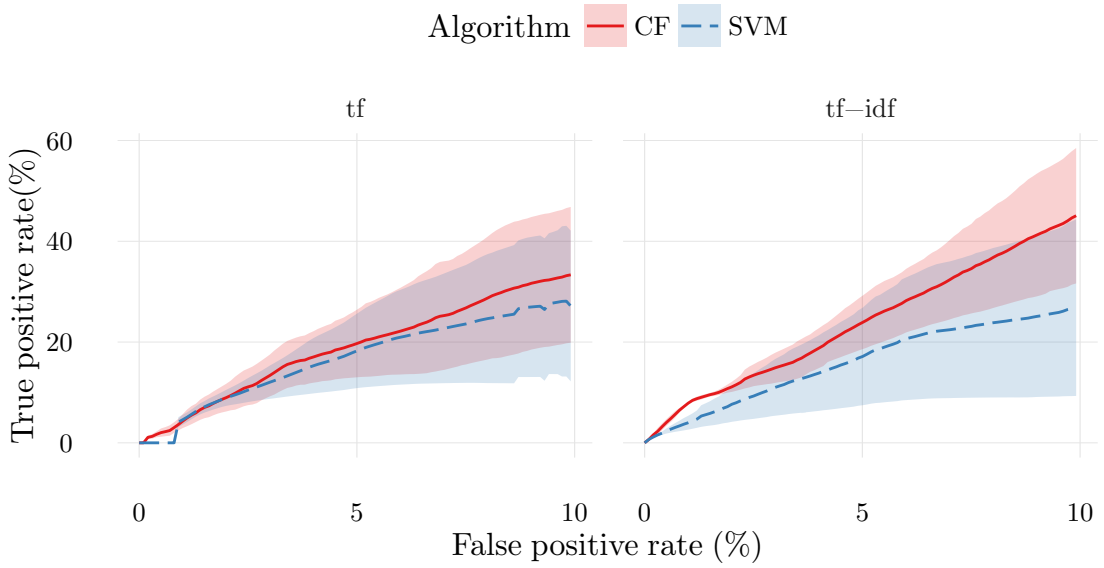


Figure 7.9: Receiver Operating Characteristic (ROC) curves comparing both algorithms. On the left, the results from comparing both algorithms when term frequency (tf) was used to create the profiles; on the right, the same comparison for term frequency – inverse document frequency (tf-idf). The bands mark the region of the standard deviation in the true positive rate for the 10 tests.

The results for tf-idf differ from the results for the tf approach. In this approach CF exhibited consistently a higher true positive rate than SVM for the same false positive rate. At a false positive rate of 0% neither algorithm exhibits any detection. However, when the false positive rate is increased to 2.5%, CF exhibits a true positive rate of $13.4 \pm 2.16\%$ while SVMs of $9.27 \pm 4.52\%$. At 5% false positive rate, the values for true positive rate for CF and SVMs are $23.87 \pm 5.30\%$ and $15.12 \pm 9.66\%$, respectively. At last, at 10% false positive rate CF exhibits a true positive rate of $45.48 \pm 13.61\%$ while SVMs of $26.97 \pm 17.66\%$.

## Final remarks

Both algorithms performed better using the Forrest method for data pre-processing, not only by exhibiting higher true positive rates but also by exhibiting significantly lower standard deviations in the true positive rate. For very small false positive rate (<2.5%) SVMs exhibit a higher true positive rate with a very small standard deviation in it. As a consequence, in that range SVMs should be used. However, if a false positive of 6.25% can be allowed then, the cellular frustration algorithm is a better choice as it exhibits a higher true positive rate even considering the standard deviation.In some problem instances the cellular frustration algorithm exhibits an anomaly detection performance that is 1.5 times the performance achieved by support vector machines.

In terms of accuracy and considering a false positive rate of (6.25%) and the parameters that give the best detection performance in each algorithm, both algorithms achieved an accuracy above 50% – $61.12 \pm 5.84\%$ and $58.78 \pm 0.03\%$ (cellular frustration algorithm and support vector machines, respectively).

One point to take into account is that the results presented here seem to contradict some of the results of the literature (lower true positive rate for a given false positive rate), for instance the ones presented in [5] and in [13]. However, we alert the reader that those results were obtained considering models trained with information from anomalies, which is not the case here. Furthermore, the results showed here don't take into account parameter optimization. That is, all results were obtained with the default parameters in both algorithms. In the case of SVMs other kernels can be more suitable. In the case of the cellular frustration algorithm, there are several parameters that can be adjusted, namely the detectors connectivity (men don't see all women - only a fraction) and the number of iterations required before changing samples $T_S$.

## 7.7 Conclusions

In this paper we described how the cellular frustration algorithm can be implemented to perform anomaly detection in the computer security field. The conducted set of experiments involving the 1999 DARPA BSM data set show that the cellular frustration algorithm in its present state can detect intrusions even when different datasets are used for training. However, the algorithm is still in development and future developments may improve its true positive rate for small false positive rates as well as reduce the true positive standard deviation.

Future work, involves benchmarking the cellular frustration algorithm in other datasets and explore if other set of parameters can increase its detection rate and decrease its standard deviation.

## 7.8 Bibliography

[1] B. F. Faria, A. Zúquete, and A. M. Lindo. Intrusion detection using the cellular frustrated framework. *(submitted)*, 2016.

[2] Hervé Debar, Marc Dacier, and Andreas Wespi. Towards a Taxonomy of Intrusion-detection Systems. *Computer Networks: The International Journal of Computer and Telecommunications Networking - Special issue on computer network security*, 31(9): 805–822, April 1999. ISSN 1389-1286.

[3] S. Mukkamala, G. Janoski, and A. Sung. Intrusion detection using neural networks and support vector machines. In *Proceedings of the 2002 International Joint Conference on Neural Networks, 2002. IJCNN '02.*, volume 2, pages 1702–1707, 2002.

[4] J. Shun and H.A. Malki. Network Intrusion Detection System Using Neural Networks. In *Fourth International Conference on Natural Computation, 2008. ICNC '08.*, volume 5, pages 242–246, Oct 2008.

[5] Wenjie Hu, Yihua Liao, and V. Rao Vemuri. Robust Anomaly Detection Using Support Vector Machines. In *In Proceedings of the International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc, 2003.

[6] Reda M. Elbasiony, Elsayed A. Sallam, Tarek E. Eltobely, and Mahmoud M. Fahmy. A hybrid network intrusion detection framework based on random forests and weighted k-means. *Ain Shams Engineering Journal*, 4(4):753–762, 2013. ISSN 2090-4479.

[7] Jiong Zhang and M. Zulkernine. Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection. In *ICC '06 IEEE International Conference on Communications 2006*, volume 5, pages 2388–2393, June 2006.

[8] V.N.P. Dao, R. Vemuri, and S.J. Templeton. Profiling Users in the UNlX OS Environment. *International ICSC Conference on Intelligent Systems and Applications*, 2000.

[9] D. Endler. Intrusion detection. Applying machine learning to Solaris audit data. In *Proceedings 14th Annual Computer Security Applications Conference*, pages 268–279. Institute of Electrical & Electronics Engineers (IEEE), 1998.

[10] Wenke Lee, Salvatore J. Stolfo, and Kui W. Mok. Mining in a Data-flow Environment: Experience in Network Intrusion Detection. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 114–124, New York, NY, USA, 1999. ACM. ISBN 1-58113-143-7.

[11] E. Eskin, Wenke Lee, and S.J. Stolfo. Modeling system calls for intrusion detection with dynamic window sizes. In *DARPA Information Survivability Conference amp; Exposition II, 2001. DISCEX '01. Proceedings*, volume 1, pages 165–175, 2001.

[12] Dae-Ki Kang, D. Fuller, and V. Honavar. Learning classifiers for misuse and anomaly detection using a bag of system calls representation. *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop, 2005.*, pages 118–125, June 2005.

[13] Yihua Liao and V.Rao Vemuri. Use of K-Nearest Neighbor classifier for intrusion detection. *Computers & Security*, 21(5):439–448, 2002. ISSN 0167-4048.

[14] Stephanie Forrest, Steven A. Hofmeyr, Anil Somayaji, and Thomas A. Longstaff. A Sense of Self for Unix Processes. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, SP '96, pages 120–128, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 0-8186-7417-2.

[15] R. Alshammari, S. Sonamthiang, M. Teimouri, and D. Riordan. Using Neuro-Fuzzy Approach to Reduce False Positive Alerts. In *Fifth Annual Conference on Communication Networks and Services Research, 2007. CNSR '07.*, pages 345–349, May 2007.

[16] J. Gomez and D. Dasgupta. Evolving fuzzy classifiers for intrusion detection. In *Proceedings of the 2002 IEEE Workshop on Information Assurance*, pages 321–323. New York: IEEE Computer Press, 2002.

[17] F. Vístulo de Abreu, E. N. M. Nolte-'Hoen, C. R. Almeida, and D. M. Davis. Cellular Frustration: A New Conceptual Framework for Understanding Cell-mediated Immune Responses. In *Proceedings of the 5th International Conference on Artificial Immune Systems*, ICARIS'06, pages 37–51, Berlin, Heidelberg, 2006. Springer-Verlag.

[18] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[19] F. Vístulo de Abreu and P. Mostardinha. Maximal frustration as an immunological principle. *Journal of The Royal Society Interface*, 6(32):321–334, 2009. ISSN 1742-5689.

[20] Bruno Filipe Faria, Patrícia Mostardinha, and Fernão Vístulo de Abreu. Can the Immune System Perform a t-Test? *PLOS ONE*, 12(1), jan 2017. doi: 10.1371/journal.pone.0169464.

[21] P. Mostardinha and F. Vístulo de Abreu. Positive and negative selection, self-nonself discrimination and the roles of costimulation and anergy. *Scientific Reports*, 2:769, oct 2012. ISSN 2045-2322.

[22] V. Vapnik and A. Lerner. Pattern Recognition using Generalized Portrait Method. *Automation and Remote Control*, 24, 1963.

[23] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, 13(7):1443–1471, July 2001. ISSN 0899-7667.

[24] David M. J. Tax and Robert P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20:1191–1199, 1999.

[25] B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, pages 582–588, 2000.

[26] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262194759.

[27] Nathalie Japkowicz, Catherine Myers, and Mark Gluck. A Novelty Detection Approach to Classification. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 518–523, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[28] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, May 2011. ISSN 2157-6904.

[29] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X.

[30] Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, and Kumar Das. The 1999 DARPA Off-line Intrusion Detection Evaluation. *Computer Networks: The International Journal of Computer and Telecommunications Networking - Special issue on recent advances in intrusion detection systems*, 34(4):579–595, October 2000. ISSN 1389-1286.

[31] MIT Lincoln Laboratory. 1999 Attack Database, 1999. URL `https://www.ll.mit.edu/ideval/docs/attackDB.html`.

[32] Matthew V. Mahoney and Philip K Chan. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. In *Recent Advances in Intrusion Detection*, pages 220–237. Springer, 2003.

[33] Tom Fawcett. An Introduction to ROC Analysis. *Pattern Recognition Letters - Special issue*, 27(8):861–874, June 2006. ISSN 0167-8655.

CHAPTER 8

# Computation of maximally frustrated populations with GPUs[1]

Many bioinspired algorithms explore emergent properties resulting from the simultaneous interaction of many agents. It is thus natural to expect that these algorithms could benefit from massive parallelism provided by graphical processing units (GPUs). However, translating algorithms onto GPU implementations is not always trivial because hardware constraints must be considered in order to achieve good performances. Here we address the repertoire education stage in a dynamical matching algorithm for intrusion detection. In this algorithm two types of agents, data presenters and detectors, engage in a complex matching dynamics forming pairs with agents of the other type. Pairs are formed anytime two agents prefer new matchings to previous matchings. The computationally hardest part of the algorithm consists in the repertoire selection stage, where detectors are selected to maximally frustrate the dynamics and minimize pairings lifetimes. In this work we describe algorithms for GPU implementations that can speed up the repertoire education stage by 85 times.

## 8.1 Introduction

Bioinspired computation is an emerging field that intrinsically requires parallel processing [2–9]. In these algorithms computations are performed by populations with many interacting agents. Most often the interesting effects arise when feedback loops induce a complex dynamics with emergent phenomena [10–12]. In these cases, the simulation of the dynamics of each individual cannot be circumvented because it cannot be decoupled

[1] chapter submitted as: B. F. Faria and F. Vístulo de Abreu. Computation of maximally frustrated populations with GPUs. *under revision*, 2016


Figure 8.1: Illustration of the stable marriages problem (SMP). In the stable marriages problem the goal is to match a set of men and women in stable marriages (pairs). Each man and woman has *fixed* traits and preferences. Two preference lists are represented for the leftmost and rightmost woman and man in the population. Men and women prioritize marriages with the highest ranked individuals and prefer to be married than alone. A stable configuration arises when no two agents prefer each other to their current match. Solving the SMP consists in finding a sequence of interactions between men and women (an agent pairing dynamics) that, for any population allows finding a stable configuration. For example the Gale-Shapley solution (also known as the ballroom dancing solution) consists in sequencially presenting each man to women until he finds a match and repeating the procedure until all men are stably married. In the cellular frustration models discussed here, the agent pairing dynamics is predefined and instead, preference lists for agents from one sex should be selected to make marriages as short lived (unstable) as possible.

from the dynamics of the remaining population. Analytical solutions are also not available except for very simple systems. The development of parallel computational algorithms is then particularly relevant as the number of agents necessary to capture the desired effects may not be small. Massive parallelism can thus be crucial, especially for practical applications, in which computational efficiency dictates methods usefulness.

The model we address in this work received inspiration from the Gale and Shapley stable marriages problem (SMP) [13–16]. The aim of the SMP is to match two sets of agents in stable pairs, so that no two agents prefer each other to their current match (see Figure 1). Finding algorithms that perform this task in polynomial time has caught considerable attention given its relevance to market design.

Similarly to the SMP, the algorithm we study here involves two sets of agents which also use preference lists - here called interaction lists or ILists [17] - to terminate and establish pairings. Likewise, in this model two agents form a new pair whenever they prefer each other relatively to their current matches. Also, as in the SMP, all agents prefer to be paired than alone. However, a fundamental difference exists between the two types of problems. In the SMP researchers look for algorithms that match agents in stable pairs.

Consequently, the agent pairing dynamics - that is, the sequence of interactions that agents should follow - is optimized to guarantee that all agents reach stable pairs. By contrast, the model discussed here looks for a set of agents that make the dynamics maximally unstable. The aim is to evaluate how much one set of agents changed by measuring how the stability of the pairings with agents of the other type was increased. Then generalized kinetic proofreading mechanisms [13, 18] amplify minor differences in the stability patterns which makes this evaluation reliable. In this process, and in contrast with the SMP, agents pairing dynamics does not change and instead it is the set of interaction lists that has to be selected to make the dynamics of the whole population unstable. The agents pairing dynamics used in this work follows a simple Monte Carlo strategy that gives, on each iteration, to all agents an opportunity to interact with an agent of the other type.

The dynamics is unstable when interactions with other agents continuously promotes the termination and creation of new pairings. In those cases, even when agents establish pairings with agents in top positions of their preference (interaction) lists, the other agent preferences must not be mutual to frustrate pairings.

In [13, 19] it was argued that this new way of looking at matching problems constitutes a general framework - the cellular frustrated framework (CFF) - to understand the adaptive immune system. The simplest model in this framework is formed by agents of two types (see Figure 8.2), corresponding to the men and women in the SMP. In the immune system these are called APCs (antigen presenting cells) and T cells. In more data mining oriented applications, they should be more appropriately called data presenters and detectors, respectively. In this model detectors can only display two different traits, while data presenters display as traits numbers from a dataset. Detectors' preferences rank all possible data numbers in a list, in decreasing order of preference. Since detectors can only display two different traits, presenters can only have two possible ILists: either they prefer to interact with one type of detectors or the other. As a result, data presenters and detectors can be divided in two subtypes, depending, respectively, on their ILists and displayed traits.

The CFF provides a new type of artificial immune system [20–24]. Artificial immune systems have been gaining increasing relevance for their wide range of potential applications in anomaly and intrusion detection [25–28]. An important achievement of this class of algorithms is that it can perform perfect self-nonself discrimination [19], i.e., distinguish the data sequences used by the system during its normal operation from other sequences. This was an important result for intrusion detection applications. It showed that it is possible to classify samples in two classes using a set of detectors that register only the two types of information implicitly.

A simple argument explains why perfect self-nonself discrimination can be achieved in cellular frustrated systems [19]. Indeed, if data presenters and detectors engage in a maximally frustrated dynamics during a repertoire education stage – i.e., in a dynamics in which pairing lifetimes are minimal – then the agent presenting the new (or nonself) data sequence engages in a less frustrated dynamics and establishes long contacts more

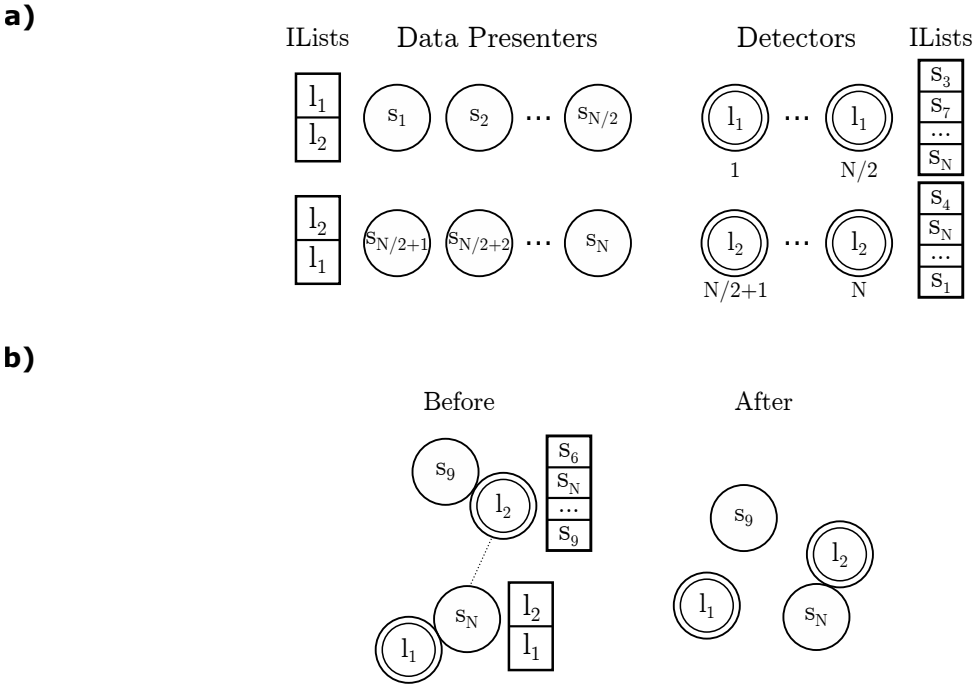


Figure 8.2: a) The representation of the model with two types of agents, data presenters and detectors. Data presenters display very diverse data sequences; detectors can only display two traits (also called ligands), $l_1$ or $l_2$, which determine the detector's subtype. All agents have preference lists (also called interaction lists, ILists) which encode information on how agents interact with ligands displayed by agents of the other type. Since detectors display only two different ligands, data presenters of the same subtype have the same IList, indicated on the left: subtype $I$ data presenters prefer ligands displayed by subtype $I$ detectors and the reverse happens with subtype $II$ data presenters. Detectors have much more complex ILists, ranking data sequences in potentially different orderings. Examples of ILists are presented on the side for one agent of each type and subtype. b) When two agents interact they take a decision that determines their next state. A new pair is established whenever both agents involved in the interaction prefer each other to the other agents they are coupled with. In that case, the other agents are freed. In this algorithm it is assumed that all agents prefer to be paired than to be alone.

frequently. To understand why this happens, consider the organization of agents in subtypes as represented in Figure 8.2. Given that subtype 1 presenters have on the top of their ILists subtype 1 detectors, education should eliminate subtype 1 detectors having on top positions sequences displayed by subtype 1 presenters as in that case they would establish long lived pairings. Education (negative selection) replaces these detectors by detectors with randomly ordered ILists. Indeed, for intrusion detection applications no a priori information should be available on how data is presented, so that no intruder could profit from this information. Therefore, ILists must be randomly drawn as for each system different presenters can present data sequences differently. Still, after many iterations in this education stage, only detectors having on top positions of their ILists sequences

presented by data presenters of the opposite subtype will remain in the population: these are the only ones that cease from being replaced. Note however that sequences that have not been presented during the education stage, will remain uniformly distributed on detectors ILists, since they never had an impact during the education. Consequently, whenever a sequence that was not presented during the repertoire education stage is presented later, in a detection stage, the data presenter displaying it will establish long pairings with a finite fraction of the detectors in the population. This explains why a dichotomy emerges in the dynamics of the system after the education process.

Importantly, in a forthcoming publication, it will be shown that cellular frustrated systems can also discriminate the absence of a combination of sequences that usually appear together. This represents a capability of detecting correlations in a sample. Together with the ability to discriminate self from nonself, it allows cellular frustrated systems to work as competent anomaly detection data miners [29].

As in any anomaly detection algorithm, two main stages can be identified. A training or repertoire education stage, during which samples defining the normal operation of the system are displayed by data presenters, and a detection stage, where other samples are presented for evaluation. The computationally limiting step in the algorithm is clearly the repertoire education stage. In populations presenting 100 different sequences it is typical that this stage requires $10^7$ evaluations per agent. This is the number of iterations typically required to decrease the maximal pairing duration in a population to a minimum value and beyond which, the algorithm cannot further decrease the maximal pairing duration. Furthermore, several ($\sim 40$) independent populations are required to build a repertoire of detectors. In contrast, the detection stage requires running the population dynamics for $10^4$ iterations ($10^4$ evaluations per agent). Beyond this number of iterations detection rates do not change. Hence, the repertoire education stage requires at least 40000 times more computational power. For this reason we address in this article how this part of the algorithm can be implemented efficiently using GPUs. Our implementation achieves computational gains of almost two orders of magnitude relatively to single CPU executions, making this new type of dynamical frustration algorithms amenable for practical applications.

This article is organized as follows. In the next section the cellular frustrated model is formally defined. Then we introduce a modified version of this model amenable to a parallelized computation. In section 8.3 we will discuss particular details on the GPU implementation strategy. In section 8.4 we will present computational performance results and compare them with equivalent CPU implementations.

## 8.2 Model

In order to make the presentation more concise we will present our model through a set of definitions. The first definition concerns the main building block: the agent. Agents can be formally defined as follows:

**Definition 8.1.** *(**Agent**) An agent $a_i \in A$ can be defined by a tuple $a_i := (s_i, l_i, K_i, R_i, p_i)$, where:*

- *$s_i \in \Sigma^b$ is an array with $b$ binary digits ($\Sigma = \{0, 1\}$), called the string or ligand;*
- *$l_i$ is an ordered sequence of all strings in $\Sigma^b$, called the list or receptor;*
- *$K_i$ is the connectivity vector holding the set of all agents that agent $a_i$ can interact with;*
- *$R_i$ is the set of interaction decision rules, which are assumed to be the same for all agents ($R_i = R, \forall a_i \in A$);*
- *$p_i \in \mathbb{N}_0$ specifies the agents pairing state and it holds the agent index to which the agent is paired, or zero if it is alone.*

Motivated by the immune system, we divide agents in two main types, data presenters (or simply, presenters) and detectors. In the immune system, presenters scavenge the body looking for antigen. They will then present antigen to $T$ cells in lymph nodes. $T$ cells work as detectors triggering an immune response if they interact with foreign antigen. In our model, presenters can potentially display any string $s_i \in \Sigma^b$. Detectors are agents that perform interactions with presenters. In particular, for intrusion detection purposes, detectors should interact differently with presenters displaying strings typically presenting during the system's normal functioning or other uncommon strings, called nonself strings. We establish the following useful definition:

**Definition 8.2.** *(**Agent Types**) The set of agents $A$ is formed by two sets (or types) of agents, presenters, $P$, and detectors, $D$, such that, $A = P \cup D$.*

For simplicity, we assume that agents of a same type do not interact. Hence, $K_i \subseteq P \forall A_i \in D$ and $K_i \subseteq D, \forall A_i \in P$. We will denote by $N_P$ the total number of presenters, i.e. $N_P \equiv \#P$, while the number of detectors is $N_D (N_D \equiv \#D)$. The total number of agents will be denoted by $N$. From Definition 8.1 it is clear that agents cannot be paired with more than one agent at a time. This constraint is respected by the set of interaction decision rules $R$ that dictate agents' pairings. Pairings depend on each agents decision which take into consideration how strings displayed by other agents are ranked in their list $l_i$. If $r_l(s)$ represents the rank of a string $s$ on list $l$, then the following decision rules can be defined:

**Definition 8.3.** *(**Decision Rules**) The following set of decision rules $R$, can be defined when agents $a_i$ and $a_j$ interact:*

- *if $p_i = 0 \wedge p_j = 0$, then $p_i \to j, p_j \to i$.*
- *if $p_i = k \wedge p_j = 0 \wedge r_{l_i}(s_j) < r_{l_i}(s_k)$, then $p_i \to j, p_j \to i, p_k \to 0$.*
- *if $p_j = k \wedge p_i = 0 \wedge r_{l_j}(s_i) < r_{l_j}(s_k)$, then $p_j \to i, p_i \to j, p_k \to 0$.*

- *if* $p_i = k \wedge p_j = m \wedge r_{l_i}(s_j) < r_{l_i}(s_k) \wedge r_{l_j}(s_i) < r_{l_j}(s_m)$ *then* $p_j \to i, p_i \to j, p_k \to 0, p_m \to 0$.

The c-Subtypes model is a generalization of the model discussed in the introduction. All data presenters or detectors of the same subtype have, respectively, the same receptor or display the same string. In order to achieve a better convergence during the education process, data presenters do not have arbitrary receptors. Rather, data presenter receptors of different subtypes are cyclically ordered, as defined below:

**Definition 8.4.** *(**c-Subtypes Model**) In a c-Subtypes model detectors and presenters are organized in c disjoint sub-sets. The set of detectors $D_I$ belonging to subtype $I = 0, \cdots, c-1$, display the string $s = I$, corresponding to the binary representation of $I$. The set of presenters $P_I$ belonging to cluster $I$ have a list $l$ whose $k^{th}$ position, $l(k)$ is given by: $l(k) = k - I + 1 (\mathrm{mod}\ c)$.*

In this paper we will only consider symmetric c-Subtypes models, for which $\#D_I = \#P_I$, $\forall I \in \{0, \cdots, c-1\}$.

**Condition 8.1.** *(**Stochastic Iteration**) Consider a sequence with all agents of a given type in the system. During a stochastic iteration all agents in this sequence are put in iteration with another randomly picked agent of the other type, and decision rules (Definition 8.3) are used to update their pairing states.*

Note that since several agents could be interacting with a same agent, some level of sequentialisation in the process is required. This will be discussed in the next section. In any case, it should be clear that our study is only concerned with the sequence of configurations that can be defined at the end of each iteration. A configuration can be defined in a precise way as follows:

**Definition 8.5.** *(**Configuration**) A configuration $C$ of the cellular frustrated system can be defined as the set of unpaired agents plus the set of pairs: $C = a_k : p_k = 0, k = 1, \cdots, N \cup (a_k, a_m) : p_k = m \wedge p_m = k, k, m = 1, \cdots, N$*

Another important concept concerns the pairing duration. This is defined as follows:

**Definition 8.6.** *(**Pairing Duration**) The duration of a pair formed by agents $a_i$ and $a_j$ is the number of iterations mediating two consecutive updates of these agents pairing states $p_i$ and $p_j$.*

In this article we explore algorithms that maximize dynamical frustration. This is done by ordering detectors lists in order to minimize pairing durations, in a negative selection process defined below:

**Definition 8.7.** *(**Negative selection with adaptive threshold**) In repertoire education with adaptive threshold, $\tau_{NS}$ is updated to the largest matching lifetime in the last $W \in \mathbb{N}$ iterations if no matching lasted longer than $\tau_{NS}$.*

Therefore, the aim of the negative selection process is find a set of detectors for which $\tau_{NS}$ is minimal.

## 8.3 GPU implementation

Graphical processing units (GPUs) have been gaining growing recognition as a new scientific tool. GPUs made the power of computer clusters accessible to a wider community. However, in order to gain these computational performances, the programmer has to incorporate knowledge of GPUs hardware architecture onto algorithms. Indeed, GPUs achieve their best performances when calculation with multiple cores profits from fast access to local memories on each core (registers), or sets of cores (shared memory), and fast context switching [30, 31]. This imposes computational implementation concerns. One is common to any parallel computation. It concerns potential conflict between simultaneous thread executions, as different threads may attempt to update the same variable. These conflicts appear easily in our model. In the next section we will discuss how these problems can be avoided.

The other problem concerns implementation strategies that must be used to profit from quick accesses to memory. Since local and shared memories are typically small, algorithms should be efficient in terms of their expenditure. In our model, registering lists associated to each agent can be memory demanding and the number of agents can be large. Therefore, two strategies were adopted. One concerned devising an implicit method for encoding agents' lists in a single integer. This will be detailed in the following subsection. The other consisted in avoiding the use of shared memory, so that the algorithm could account for sufficiently large populations.

Finally, another task that required a GPU implementation consisted in producing a random permutation of numbers to put agents in interaction at each iteration. For this we used the Radix_sort and algorithms available in the literature [32, 33].

### Conflicting Decisions Avoidance

In our model, configurations change as agents continuously attempt to pair with agents that are ranked higher in their lists. An algorithm establishing how interactions and decision rules are applied must be consistent with the fact that in each configuration agents can only be connected to a single agent at each time. This is not trivially guaranteed when contacts among agents take place in parallel. In Figure 8.3, we show that if the parallelized algorithm is not carefully designed, an agent $a_i$ could be paired with two different agents in the next configuration.

These conflicting decisions on the state of an agent arise when $a_i$ attempts to pair with agent $a_k$, while agent $a_m$ attempts to pair with $a_i$. In that case, an independent application of decisions rules could pair agent $a_i$ with two different agents in the next configuration as illustrated on the left in Figure 8.3. Another example of conflicting

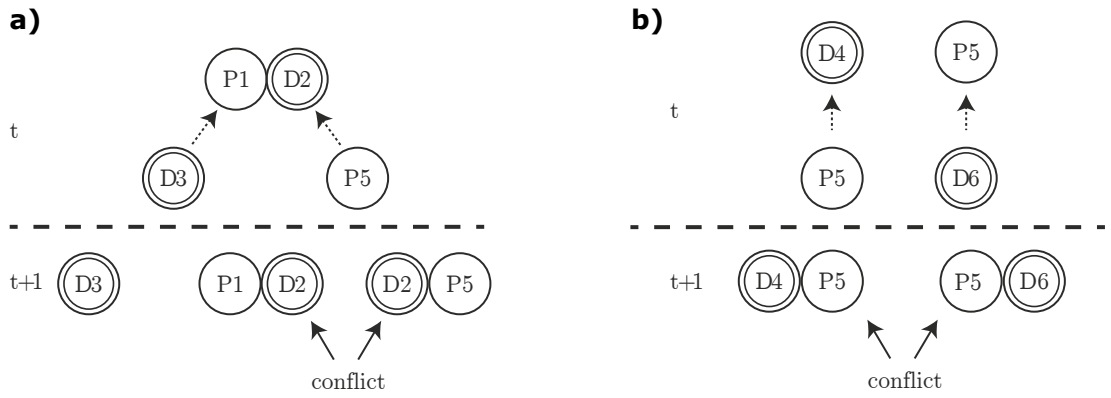


Figure 8.3: Typical inconsistencies generated by conflicting decisions that could emerge when multiples cores perform independent computations. For an efficient parallelization conflicting decisions must be avoided. **a)** If a core computes the decision of presenter agent $P1$ and another one of detector agent $D2$, then the first core could predict that agent $P1$ continues paired to $D2$, while the other core could pair agent $D2$ to $P5$. The next configuration would associate two different states to agent $D2$, which is inconsistent with the constraint that forces agents to be matched with a single agent at a time. **b)** A similar inconsistency emerging with unpaired agents.

decisions on the state of an agent could take place when two paired agents interact with two other agents. If one of these interactions leads to a change in the pairing state, while the other does not, then again a same agent could be paired to two different agents, as shown on the right in Figure 8.3.

These conflicting decisions occur because interactions are executed in parallel and independently. In principle, solving these conflicts could imply comparing information on a scattered set of processors. Eventually it could require having access to the same memory location by different processors. This would introduce delays and consequent performance loss. To avoid this, one should develop algorithms that use data parallelism. This means that all processors should use the same information to compute independent outputs that are straightforwardly forwarded onto the next configuration. Our strategy for achieving data parallelism consists in forcing that each agent makes a single decision on each iteration step. This is done by establishing that all agents of a given type (e.g., presenters) interact with a random permutation of agents of the other type. Each thread computes the decision resulting from an interaction between two agents. Since each agent interacts only with one other agent, the result of the computation of each thread establishes whether the two agents involved in the interaction will form a pair or not. If they do, then this information should be outputted (Figure 8.4). Otherwise, there are two possible fates for the agents involved in this interaction. If they were not paired, then they will remain unpaired. If they were paired, then their state can still be changed, depending on the interaction between the agent they were paired with, and the agent of the other type that agent is interacting with. Two outcomes will then be possible. If

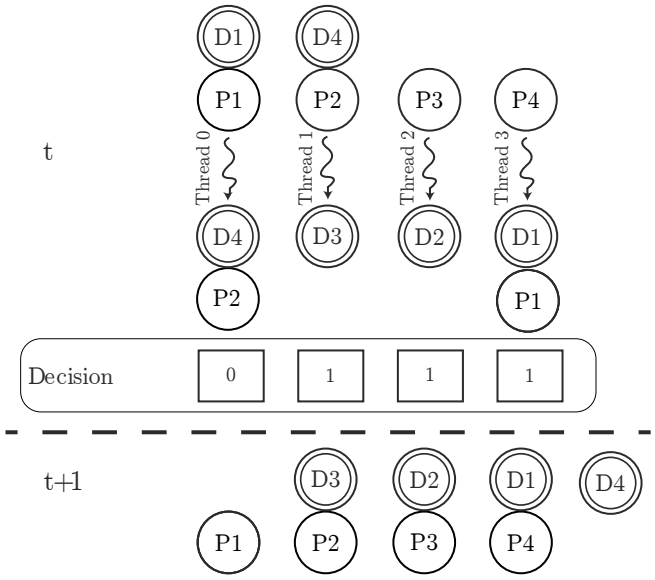


Figure 8.4: To avoid inconsistencies and to allow independent thread execution, presenters (denoted by $P$ in the figure) interact with a random permutation of detectors (denoted by $D$) using different threads. For instance, presenter agent $P1$ which is paired with $D1$, is put in interaction with agent $D4$, which is paired with agent $P2$. Each thread then applies decision rules and computes a decision. In this way each agent performs a single decision in each iteration. If they agree to establish a new pair then this establishes the state of the agents involved in this decision immediately. This is the case of agents $D3$ and $P2$, $P3$ and $D2$, or $P4$ and $D1$. If this does not happen (as in the interaction between $P1$ and $D4$) then this decision has no impact in the next configuration and consequently the pairs involved ($D1P1$ or $D4P2$) can only terminate through an interaction involving the agents to which $P1$ or $D4$ are paired.

those agents decide to form a new pair, the former agent will be left alone. Otherwise it will remain in the previous state. In any case, the output of this thread is independent from the other thread. As a result the outputted information from the two threads can be used to establish the state of all agents involved in the next generation, as illustrated in Figure 8.4.

To summarize, in this algorithm, all threads are executed independently, and their outputs change different variables. The next configuration can be computed using the information produced by the independent computation of all threads, setting the stage to compute a new iteration.

## Implicit List Definition

In the cellular frustration model a list is associated to each agent where all the possible strings presented by agents of the other type are ranked. The important point is that ILists can rank strings in completely arbitrary orderings. Storing lists is however unpractical for two reasons. First because the amount of memory required would grow exponentially

with string length. Secondly, the computational time required to search a string in the list would also grow with the list size. Our implementation uses an implicit definition of lists. Instead of generating lists explicitly, each agent is associated a seed number and a method to compute a score $\sigma$ based on the string $s$ presented by other agents. Hence, agents have a method to compute scores for any string, and consequently they could order them in a list, if required.

The method consists in applying sequentially one of two random number generators, $G_0$ or $G_1$, depending on the sequence of $b$ bits on the string $s$ presented by the other agent. This algorithm can be written in mathematical form as:

$$\sigma_i = G_1(\sigma_i)^n \delta_{k_i,1} + G_0(\sigma_i)^n \delta_{k_i,0} \tag{8.1}$$

where $i = 1, \cdots, b$ and $n$ is an integer. $\sigma_1$ is assumed to be given and $\delta_{ij}$ is the Kronecker symbol. Also, $k_i = (\lfloor \frac{s}{2^i} \rfloor$ AND 1), gives 0 or 1 depending on whether the $i^{th}$ digit of the $s$ string is 0 or 1, respectively (here $\lfloor \cdot \rfloor$ represents truncation to integers).

For a matter of illustration, consider that the string read was $s = 011$. Then the method computes a score according to $G_0(G_1(G_1(\sigma_1)^n)^n)^n$: it applies $G_1$ to a seed number $\sigma_1$, $2n$ times – $n$ times for each bit equal to 1 – and afterwards $G_0$ $n$ times. Finally, $G_0$ and $G_1$ are two different random generators. We chose two linear congruential generators (LCG):

$$G_j : \sigma(i+1) = a_j \sigma_i + c_j (\text{mod } m), j = 0, 1 \tag{8.2}$$

with $m = 2^{32}$,$a_0 = 134775813$, $c_0 = 1$ and $a_1 = 214013$, $c_1 = 2531011$. These parameters have been used in several commercial compilers, like Borland Delphi and Microsoft Visual C. The important point is that all arbitrarily ordered lists can be generated using this procedure. We tested that this was indeed the case for systems with 8 agents displaying all possible strings with 3 bits. We verified that for $n > 1$, all $8! = 40320$ lists could be generated, by changing the seed number $\sigma_1$. We also confirmed that for systems with 9 agents and 4 bits, all $9! = 362880$ can be generated. Hence, this method can efficiently encode all agents' lists. In practice, for a system with 512 presenters and 512 detectors, registering lists explicitly requires storing $512 \times 2 = 1024$ arrays with 512 integers each, i.e., $5 \times 10^5$ numbers. This should be compared with storing only 1024 seed numbers. This memory saving strategy is clearly important in GPU computations as local memories allow fast accesses. However, as they are a limited resource it is crucial to devise strategies to optimize them.

Finally, it should be mentioned that the global memory access pattern in GPUs plays a determinant role in algorithms computational performances because global memory accesses are slow. Optimal performances are achieved when threads in a warp access consecutive memory regions. In that case, threads memory access can be coalesced into a single request. By contrast, when neighbouring threads access scattered memory locations, then memory accesses cannot be coalesced and computational performance decreases. Since, accessing slow memory severely affects computational performance, it is often

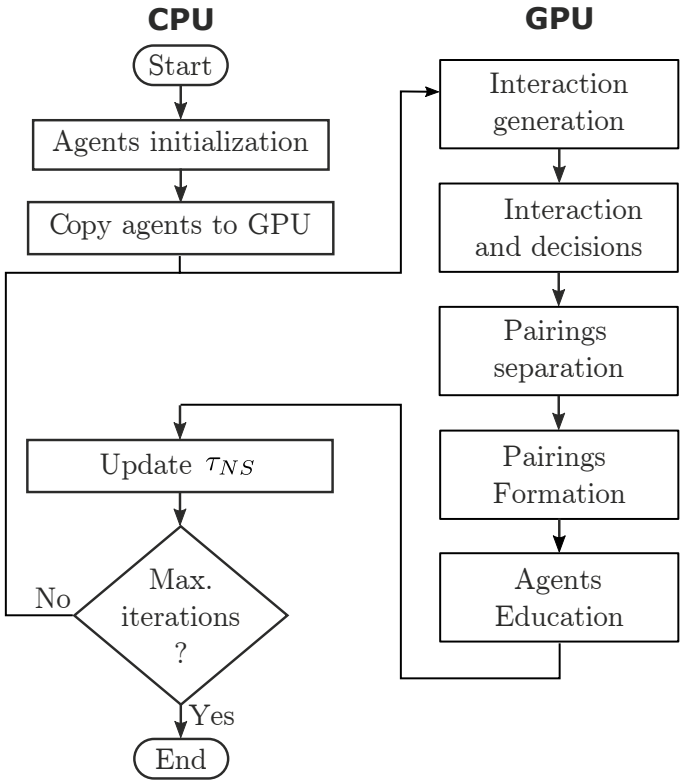


Figure 8.5: The overall algorithm for computation of maximally frustrated populations. On the left are displayed tasks executed by the CPU, while on the right are those executed by the GPU device.

preferable to recalculate certain quantities rather than storing them in advance, for later use. Therefore, only certain quantities like presenters ligands and detectors receptors ($s_i$ and $p_i$, respectively) and their pairing states $p_i$, are stored in global memory, while presenters receptors and detectors ligands are computed in a per thread basis.

## Algorithm for Computing a Maximally Frustrated Population

The overall algorithm is presented in Figure 8.5. The flowchart shows on the left side tasks performed in the CPU and on the right, those performed in the GPU. The algorithm uses the CPU to initialize the program and to launch kernels in the device. The main steps are the following. First all agents are initialized and transferred to the GPU. Then the GPU computes a random permutation of agents' indices and establishes which agents of different types interact. Then, decision rules are applied. Each thread computes the decision associated to two interacting agents of different types. The determination of the next configuration follows two steps. First threads with agents willing to change pair are separated from previous pairings. Since not all threads participate in this task, a synchronization point is set at this stage. Then state variables associated to agents forming new pairs are updated, i.e., new pairs are formed. Again a thread synchronization

point takes place at this stage.

The next step is to perform education on the detectors repertoire. In this step all pairs that have been formed more than $\tau_{NS}$ iterations ago are separated, and the detector's seed number generator is replaced by a new one.

On the CPU, the threshold value, $\tau_{NS}$, can be periodically updated after a fixed number of iterations, $W$. Several strategies for updating $\tau_{NS}$ can be used. One of the simplest consists in using a fixed and sufficiently small value for $\tau_{NS}$. This has nevertheless the disadvantage of requiring a prior knowledge of what a good value could be. Alternatively, $\tau_{NS}$ can be updated when no pairing durations exceeded $\tau_{NS}$ in the last $W$ iterations. This was the method we used to obtain the results reported in this article. In that case $\tau_{NS}$ is set equal to the largest pairing duration registered during those iterations. Each time $\tau_{NS}$ is changed the population is recorded.

## 8.4 Results

We will discuss two types of results. In the next section we show that frustration is maximized along the education process. Afterwards we will compare computational performances using our GPU strategy and equivalent serial and parallel programming implementations.

### Frustration Maximisation

Frustration is maximized when pairing durations are minimized. The negative selection algorithm progressively replaces detectors performing the longest pairings. In Figure 8.6 the distributions for the full set of presenters are presented in the initial (Figure 8.6-a)) and the final iteration (Figure 8.6-b)). It is clear that pairing durations are considerably reduced after education. These plots were obtained with a small modification on the education algorithm discussed in the previous section: instead of performing selection and updating $\tau_{NS}$, a record of the number of pairings of a given duration for each agent is continuously updated in the GPU, and is sent to the CPU and saved in the last iteration. These results are similar to those already reported in [19]. However, now two main differences exist in the algorithm. First, parallelization required establishing that a different way of selecting agents for interaction and then agents' lists were defined implicitly. Hence, these results confirm that similar results can be obtained with this more efficient implementation of the algorithm.

In Figure 8.6 frequencies for the pairing durations of the whole population are also plotted. These graphs are also plotted for a sequence of iterations, being represented in lighter grey in the initial iterations (Figure 8.6-c)) and in darker grey for the final iterations. It is clear from these results that pairing durations are considerably reduced for all detectors in the population. Finally, in Figure 8.6-d) we show how the largest pairing duration evolved along the education process for 100 populations. With the increase in

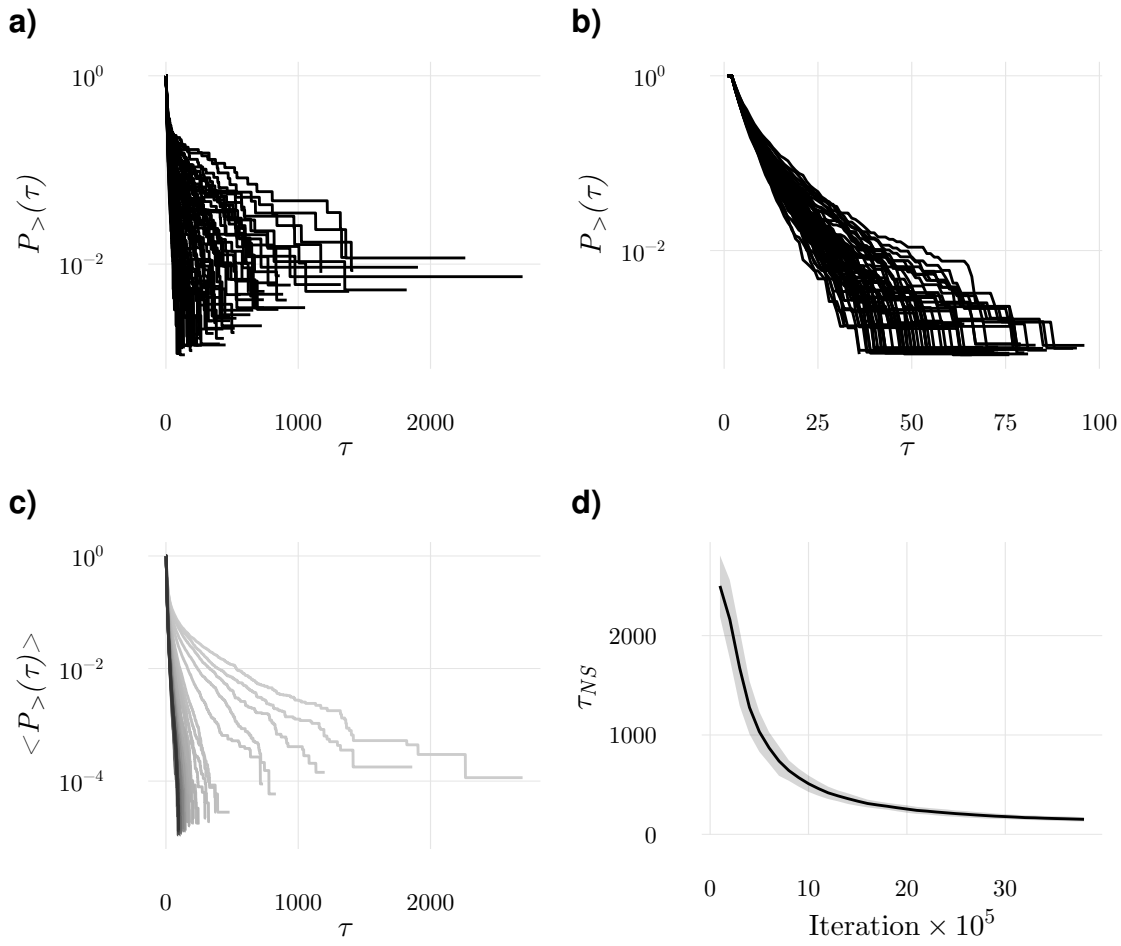


Figure 8.6: Number of pairings lasting at least $\tau$ iterations for each presenter in a population with 64 agents of each type, $P_>(\tau)$, at the beginning a) and in the end of the education process b), for simulations with 10000 iterations. The education process took $4 \times 10^6$ iterations. c) Average number of pairings lasting at least $\tau$ iterations taking into account all presenters in a population, and at several stages of the education stage. The line with the largest number of longest pairings corresponds to the average of all curves in a). The darkest line corresponds to the average of all curves in b). The other lines correspond to results obtained with populations educated with an increasing number of iterations. d) The average of the largest pairing duration registered in 100 populations in the last $\tau_{NS}$ update in the education stage. The grey region accounts for $\pm$ the standard deviation. Hence, for instance, when populations that have undergone $10^6$ iterations, the largest pairing registered in the last 10000 iterations is 500. Taken together these results show the convergence of the selection progress, reducing significantly the largest pairing durations.

the number of iterations the largest pairing duration decreases to a minimal value. The distribution of pairing durations is also much narrower in the end of the education process than in the initial iterations. Hence education has two effects: it reduces pairing durations and makes them more uniformly distributed. It should be remarked that in the results in

Figure 8.6-d) all populations presented different sets of strings. Hence, even though there is considerable diversity in the populations, their dynamics is analogous.

## Computational Performance

Our test platform was composed of an Intel core i7 (CPU) with 24GB of RAM (DDR3), and one graphic card dedicated to computation. The graphic card used was the NVIDIA Geforce GTX-580 with 1.5GB of RAM (GDDR5). In order to provide a good assessment of the computation performance, the same algorithm was implemented in both CPU and GPU. Two cases were considered for the CPU implementation. The first implementation educated one population at a time and consequently it used only one core. The second implementation took advantage of all the CPU resources by educating several populations at a time (multi core). In this case, given that the maximum number of cores was 6, 12 considering hyper-threading, we performed simultaneous independent education of 12 populations. To compare computational performances we calculated the amount of time required to educate 1000 populations of detectors for 10000 iterations (Figure 8.7). We considered populations with different sizes. To take full advantage of the GPU architecture we considered populations with a number of agents that is a multiple of 32. In the present study we considered populations with a total numbers of agents of 128, 256, 512, 1024, 2048 and 4096. The total amount of time required for the several populations is shown in Figure 8.7-a), while the computational gain (or speed up) given by the increase in computational times required by the CPU implementations relative to the GPU implementation is shown in Figure 8.7-b).

Our GPU implementation reduced significantly the computational time required. The biggest gains were obtained for the biggest systems. When the total number of agents is 4096, gains topped to 85 times. However, even for small populations (with 64 presenters and 64 detectors) gains of $\sim 60$ times were achieved. In contrast, when the multi core CPU implementation was used the GPU computational gains reduced to 6.9 and 9.3 times, respectively. The remarkable feature of using GPUs regards the increase in the computational effort when the number of agents is increased. For instance, for the task considered, $64 + 64$ agents required $5.25s$ while $4096 + 4096$ agents required $301.75s$. That is, when the number of agents increases 64 times, the computational effort of the GPU only increases $\sim 57$ times. This is certainly quite different from what happens with conventional CPU processing for which the required computational effort grows even faster than the number of agents. For the previous example of a 64 times increase in the number of agents, the computational effort required by both CPU implementations increased $\sim 77$ times.

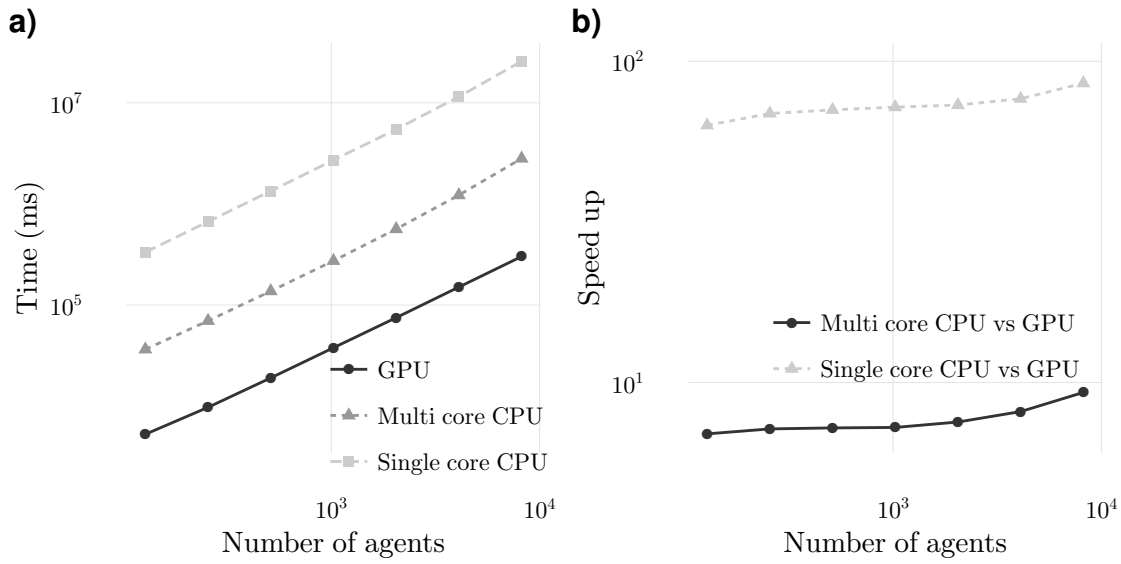


Figure 8.7: a) The computational time taken by the CPU and the GPU to educate 1000 populations of detectors with increasing number of agents for 10000 iterations. b) The speed up comparison for two CPU implementations (single core and multi core) relative to the GPU implementation.

## 8.5 Conclusions and Perspectives

In this article we showed how massive parallelism can be used to reduce considerably the time required for computing maximally frustrated populations. As discussed in [34], the computation of maximally frustrated populations is the most demanding task required for designing practical applications of the cellular frustration algorithm for intrusion detection. With current computers and algorithms this task may require a few minutes even for relatively small populations ($100 + 100$ agents). In this article we achieved speed ups of almost two orders of magnitude, which allows tackling large populations within acceptable computational times. To parallelize the algorithm the original model had to be slightly changed. It had to be assumed that all agents of one type interacted with a different agent of the other type. This, however, did not change the dynamics nor the convergence of the algorithm. Another type of strategy used to speed the algorithm had in mind that often calculations are faster than operations involving memory accesses. Therefore, instead of storing big lists of randomly permuted integers - the interaction or preference lists present also in optimization problems like the stable marriage problem - the ranking of two elements in a list can be compared by calculating scores with a random number generator. This only requires storing seed numbers and performing the calculation whenever necessary, instead of looking for items in a list.

Cellular frustrated systems are only now finding practical applications. Given the degree of novelty associated to the ideas in this class of models, and the ideas motivating them (inspired in how matchings are established in a human society or on how the immune system may work), it is plausible that this class of models can be greatly extended to

incorporate new features, address other points of view and lead to the finding of new computational capabilities. Given the numerous feedbacks in these models, their behaviour is also difficult to predict without computational investigation. As interactions in these models occur naturally in parallel, finding strategies that take full advantage of the current parallel computational power should be always of interest for researchers in this and other closely related fields[35–38].

## 8.6 Bibliography

[1] B. F. Faria and F. Vístulo de Abreu. Computation of maximally frustrated populations with GPUs. *under revision*, 2016.

[2] Jayram Moorkanikara Nageswaran, Nikil Dutt, Jeffrey L. Krichmar, Alex Nicolau, and Alex Veidenbaum. Efficient simulation of large-scale spiking neural networks using CUDA graphics processors. In *Proceedings of the 2009 international joint conference on Neural Networks*, IJCNN'09, pages 3201–3208, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-3549-4.

[3] Simon Harding and Wolfgang Banzhaf. Fast genetic programming on GPUs. In *Proceedings of the 10th European conference on Genetic programming*, EuroGP'07, pages 90–101, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-71602-0.

[4] Denis Robilliard, Virginie Marion-Poty, and Cyril Fonlupt. Genetic programming on graphics processing units. *Genetic Programming and Evolvable Machines*, 10(4): 447–471, December 2009. ISSN 1389-2576. doi: 10.1007/s10710-009-9092-3.

[5] You Zhou and Ying Tan. GPU-based parallel particle swarm optimization. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 1493–1500, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2958-5.

[6] Lucas De P. Veronese and Renato A. Krohling. Swarm's flight: accelerating the particles using C-CUDA. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 3264–3270, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2958-5.

[7] Marjan Rouhipour, Peter J. Bentley, and Hooman Shayani. Fast bio-inspired computation using a GPU-based systemic computer. *Parallel Computing*, 36(10-11):591–617, October 2010. ISSN 0167-8191. doi: 10.1016/j.parco.2010.07.004.

[8] Michael J. Hallock, John E. Stone, Elijah Roberts, Corey Fry, and Zaida Luthey-Schulten. Simulation of reaction diffusion processes over biologically relevant size and time scales using multi-GPU workstations. *Parallel Computing*, 40(5-6):86–99, 2014. ISSN 0167-8191.

[9] Sujatha R. Upadhyaya. Parallel Approaches to Machine learning-A Comprehensive Survey. *Journal Parallel Distribributed Computing*, 73(3):284–292, March 2013. ISSN 0743-7315. doi: 10.1016/j.jpdc.2012.11.001.

[10] H. J. Chiel and R. D. Beer. The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. *Trends Neurosci*, 20(12): 553–557, December 1997. ISSN 0166-2236.

[11] Dante R. Chialvo. Emergent complex neural dynamics. *Nature Physics*, 6(10): 744–750, October 2010. ISSN 1745-2473. doi: 10.1038/nphys1803. URL `http://dx.doi.org/10.1038/nphys1803`.

[12] Béla Suki, Jason H.T. Bates, and Urs Frey. *Complexity and Emergent Phenomena*. John Wiley & Sons, Inc., 2011. ISBN 9780470650714. doi: 10.1002/cphy.c100022.

[13] F. Vístulo de Abreu, E. N. M. Nolte-'Hoen, C. R. Almeida, and D. M. Davis. Cellular Frustration: A New Conceptual Framework for Understanding Cell-mediated Immune Responses. In *Proceedings of the 5th International Conference on Artificial Immune Systems*, ICARIS'06, pages 37–51, Berlin, Heidelberg, 2006. Springer-Verlag.

[14] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[15] NobelPrize.org. The Prize in Economic Sciences 2012 - Advanced Information, January 2013. URL `http://www.nobelprize.org/nobel_prizes/economics/laureates/2012/advanced.html`.

[16] C.R. Almeida and F.V. de Abreu. Dynamical instabilities lead to sympatric speciation. *Evolutionary Ecology Research*, 5(5):739–757, 2003.

[17] F. Vístulo de Abreu and P. Mostardinha. Maximal frustration as an immunological principle. *Journal of The Royal Society Interface*, 6(32):321–334, 2009. ISSN 1742-5689.

[18] André M. Lindo, Bruno F. Faria, and Fernão V. de Abreu. Tunable kinetic proofreading in a model with molecular frustration. *Theory in Biosciences*, 131(2):77–84, 2012. ISSN 1611-7530.

[19] P. Mostardinha and F. Vístulo de Abreu. Positive and negative selection, self-nonself discrimination and the roles of costimulation and anergy. *Scientific Reports*, 2:769, oct 2012. ISSN 2045-2322.

[20] Stephanie Forrest, Alan S. Perelson, Lawrence Allen, and Rajesh Cherukuri. Self-Nonself Discrimination in a Computer. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, SP '94, pages 202–212, Washington, DC, USA, 1994. IEEE Computer Society.

[21] Leandro Nunes De Castro and Fernando J. Von Zuben. The Clonal Selection Algorithm with Engineering Applications. In *In GECCO 2002 - Workshop Proceedings*, pages 36–37. Morgan Kaufmann, 2002.

[22] V. Cutello and G. Nicosia. The clonal selection principle for in silico and in vitro computing. *Recent developments in biologically inspired computing*, pages 104–146, 2004.

[23] L.N. de Castro and F.J. Von Zuben. Learning and optimization using the clonal selection principle. *Evolutionary Computation, IEEE Transactions on*, 6(3):239–251, jun 2002. ISSN 1089-778X. doi: 10.1109/TEVC.2002.1011539.

[24] Jon Timmis, Mark Neal, and John Hunt. An artificial immune system for data analysis. *Biosystems*, 55(1-3):143–150, 2000. ISSN 0303-2647. doi: 10.1016/S0303-2647(99)00092-1.

[25] Stephanie Forrest, Steven Hofmeyr, and Anil Somayaji. The Evolution of System-Call Monitoring. In *2008 Annual Computer Security Applications Conference (ACSAC)*. Institute of Electrical & Electronics Engineers (IEEE), dec 2008.

[26] Thomas Knight and Jon Timmis. AINE: An Immunological Approach to Data Mining. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 297–304, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1119-8.

[27] Julie Greensmith, Uwe Aickelin, and Steve Cayzer. *Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection*, pages 153–167. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-31875-0.

[28] Thomas Stibor, Robert Oates, Graham Kendall, and Jonathan M. Garibaldi. Geometrical insights into the dendritic cell algorithm. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, GECCO '09, pages 1275–1282, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-325-9. doi: 10.1145/1569901.1570072.

[29] Bruno Filipe Faria, Patrícia Mostardinha, and Fernão Vístulo de Abreu. Can the Immune System Perform a t-Test? *PLOS ONE*, 12(1), jan 2017. doi: 10.1371/journal.pone.0169464.

[30] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2010.

[31] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley Professional, 1st edition, July 2010. ISBN 0131387685.

[32] Jared Hoberock and Nathan Bell. Thrust: A Parallel Template Library, 2010. URL `http://www.meganewtons.com/`. Version 1.3.0.

[33] NVIDIA. *CUDA CURAND Library*. NVIDIA Corporation, Santa Clara, CA, USA, August 2010.

[34] Patrícia Mostardinha, Bruno Filipe Faria, André Zúquete, and Fernão Vístulo Abreu. A Negative Selection Approach to Intrusion Detection. In *The 11th International Conference on Artificial Immune Systems (ICARIS 2012)*, volume LNCS 7597, Taormina, Italy, August 2012.

[35] Alberto Cano, Juan Luis Olmo, and Sebastián Ventura. Parallel multi-objective Ant Programming for classification using {GPUs}. *Journal of Parallel and Distributed Computing*, 73(6):713–728, 2013. ISSN 0743-7315. doi: http://dx.doi.org/10.1016/j.jpdc.2013.01.017.

[36] G.A. Papakostas, K.I. Diamantaras, and T. Papadimitriou. Parallel pattern classification utilizing GPU-based kernelized Slackmin algorithm. *Journal of Parallel and Distributed Computing*, 99:90–99, 2017. ISSN 0743-7315.

[37] E. Calore, A. Gabbana, J. Kraus, E. Pellegrini, S.F. Schifano, and R. Tripiccione. Massively parallel lattice-Boltzmann codes on large {GPU} clusters. *Parallel Computing*, 58:1–24, 2016. ISSN 0167-8191.

[38] Ketan Date and Rakesh Nagi. GPU-accelerated Hungarian algorithms for the Linear Assignment Problem. *Parallel Computing*, 57:52–72, 2016. ISSN 0167-8191.

CHAPTER 9

# Conclusions and perspectives

The self-nonself dichotomy is still an interesting and relevant view on how the immune system performs pathogen detection. It is, however, quite difficult to think that the immune response is solely based on this dichotomy. Indeed, in the last decades several researchers brought the self-nonself model into question by proposing different views on how the immune system detects foreign elements. For instance, Polly Matzinger suggested that immune responses are not due to the presence of "nonself", but to the emission, of "danger" or "alarm" signals released by the body's own cells. Although not as drastic as Matzinger Danger theory, Grossman hypothesized that immune responses are due to the presence of nonself in a context. Within Grossman view, cells activate if the applied stimuli supersedes the activation threshold which accounts for the cell stimuli history, the context, plus a critical value. None of the proposed models has however gained the same widely acceptance as the self-nonself model, even-though it is currently thought that an immune response is not only characterized by the self-nonself discrimination. Nonetheless, these models are part of the current state of art and served as inspiration for the development of a series of successful artificial intelligence algorithms, namely the dendritic cell algorithm and some tunable activation thresholds algorithms. In this work, however, I pursued a different route. Instead of focusing on the more established and discussed immunological views to develop an anomaly detection algorithm, I focused on the cellular frustration view, proposed by the supervisor of this thesis and which has been almost disregarded by the artificial immune system community. Contrarily to the more discussed immunological views, i.e. self-nonself, Danger and Tunable activation thresholds, the cellular frustration model can describe both the self-nonself discrimination and the self-abnormal self discrimination within the same concept. The self-abnormal self discrimination was previously suggested by Mostardinha in her PhD thesis and was corroborated by the work conducted on Chapter 5 of this thesis, were it was hypothesized and verified that the cellular frustration model could perform very similar to the more widely known statistical tests, t and KS, on location problems and was comparable to the widely known data-mining algorithms, SVM and RF, for more complex problems.

Chapter 6 of this thesis focused on modifying the cellular frustration model for anomaly detection tasks. In this chapter it was shown that some of the immunological concepts (i.e. education) could be relaxed without inputting a penalty on the anomaly detection performance of the algorithm. It was observed that the proposed changes improved the computational efficiency of the algorithm by an order of magnitude. However, more importantly they have addressed one of my main concerns, i.e. the high number of model parameters that need to be specified. The other being computational efficiency. On closer inspection it was observed that the algorithm detection precision had become almost independent, at least in the tested datasets, of the specified parameters. i.e. changing the parameters had little impact on detection precision. It was observed that even-though changing the parameters can impact detection accuracy for the presentation of a single nonself ligand, the same is not true when multiple perturbations are presented at the same time.

Chapter 7 was dedicated to study the viability of using the Cellular Frustration Algorithm (CFA) in host based intrusion detection. In this chapter two different strategies to describe program behaviour, Forrest sequence of system calls and Liao system call frequency, were analysed and implemented from a semi-supervised point of view. The anomaly detection performance of the CFA was accessed by comparison with detection performance of the one-class SVM. It was observed that in semi-supervised anomaly detection both CFA and one-class SVM achieved better anomaly detection results with Forrest's approach than with Liaos strategy. When compared between each-other it was observed that the anomaly detection performance of the CFA only surpassed the anomaly detection performance of one-class SVM for a false positive rate greater than 6.5%. If a false positive rate of 10% is considered to be manageable, then the CFA represents a better alternative to semi-supervised host based intrusion detection than one-class SVMs achieving a higher number of anomaly detections.

Chapter 8 was dedicated to the analysis of new computational paradigms. In this chapter the use of Graphical Processing Units (GPUs) was investigated to reduce the cellular frustration algorithm computational time. The proposed parallel implementation not only reduced the computational time by at least one order of magnitude but it does so without compromising the model dynamical properties.

The work conducted on this thesis sought to materialize the Cellular Frustration Model (CFM) in a semi-supervised anomaly detection algorithm and apply it to intrusion detection. However, the work is not complete and several directions of research are now open. For instance, can the CFA be adapted to perform supervised anomaly detection or even classification? Several ideas come to mind to overcome this problem. Possibly one of the easiest ideas is to build the detector information mapping such that the delimited rare ligand regions contemplate an higher number of rare ligands for the anomaly training examples than the normal training examples. An argument favouring this idea is that detection in the CFA is accomplished by either: (i) a number of ligands exhibited on a detector list top positions being absent or (ii) a number of ligands, even-though not

necessarily the same, missing in a number of detector list top positions. Nonetheless, there is also evidence that this idea should not work. As discussed in the introduction of this thesis anomalies need not to follow specific assumptions. Even if this idea works for the type of anomalies presented during training, there is no guarantee that it will also work for different types of anomalies. On a closer inspection following this idea implies detection criteria (ii) to be completely overlooked as one is just increasing the number of missing ligands for each detector. Another possibility using the same idea would be to only use the mapping strategy on some detector populations. Nonetheless, even in this case there is no indication that anomaly detection performance might be increased.

A second direction of research is to describe the CFA from a mathematical point of view. From the detection perspective the required steps have already been accomplished in the uncovering of the detection mechanism present on the CFA dynamics. As previously mentioned CFA education privileges combinations of ligands that frequently co-occur to the detectors top list positions. Consequently, detection in the CFA can be described using a mathematical rule. During this thesis I developed a first approximation to describe detection by a mathematical rule. The developed approximation considers that presenters dynamical response is given by the weighted average position that the presenter exhibited ligand occupies in the lists of the detectors of the same subtype. Mathematically, in the formalism of chapter 4, the $i^{th}$ presenter response to sample $s_j$ can be described as:

$$R^i_{s_j} = \frac{1}{n}\sum_k \left(\frac{N - r_{L_k}(s^i_j)}{N}\right)^{\alpha}, \forall k : r_{L_i}(l_k) = 1 \tag{9.1}$$

where $n$ denotes the number of detectors that rank the presenter sample exhibited ligand $s^i_j$; $N$ the length of the $k^{th}$ detector receptor $L_k$; $r_{L_k}$ the rank (position) of the $s^i_j$ ligand on the detector receptor $L_k$; $r_{L_i}$ the rank of the detector $k$ on the presenter receptor and $\alpha$ a value between 0 and 1. Some preliminary results using this expression on the datasets of chapter 6 can be appreciated in the appendix A. In most datasets the developed mathematical rule exhibited comparable detection results to the ones exhibited by the CFA with agent dynamics. Nonetheless, only future work will corroborate if the developed expression is an accurate replacement for the CFA monitoring stage and if it can be used to develop an even faster education stage.

On a more personal note, I think that the algorithm developed in thesis is capable of performing more than semi-supervised anomaly detection, it could even be used to perform classification. However, I don't believe that this will be accomplished using the agent dynamics. One can argue that agent dynamics are the realization of the theoretical framework. Nonetheless, and not denying agent dynamics usefulness to uncover the detection mechanisms, any algorithm to be of practical use must be computationally efficient. Much work has been developed in this thesis towards an efficient algorithm. In spite of this effort, I believe that only a complete mathematical description of the mechanisms – possibly by making use of the detection rule – will make the algorithm computationally efficient. Sure, one can argue that there is no proof that the cellular frustration model is correct, i.e. in that it correctly describes how the adaptive immune

system works, and that further developments of this model is a waste of time, however the famous quote by George E. P. Box:

> "All models are wrong, but some are useful." (George E. P. Box)

could not make more sense, and which essentially says that models should not be categorized by correctness, because models only describe a tiny portion of nature, but by how useful they are. In this sense, only the continuous application and developments of the cellular frustration algorithm will reveal its usefulness as a model.

APPENDIX A

# Auxiliary graphs

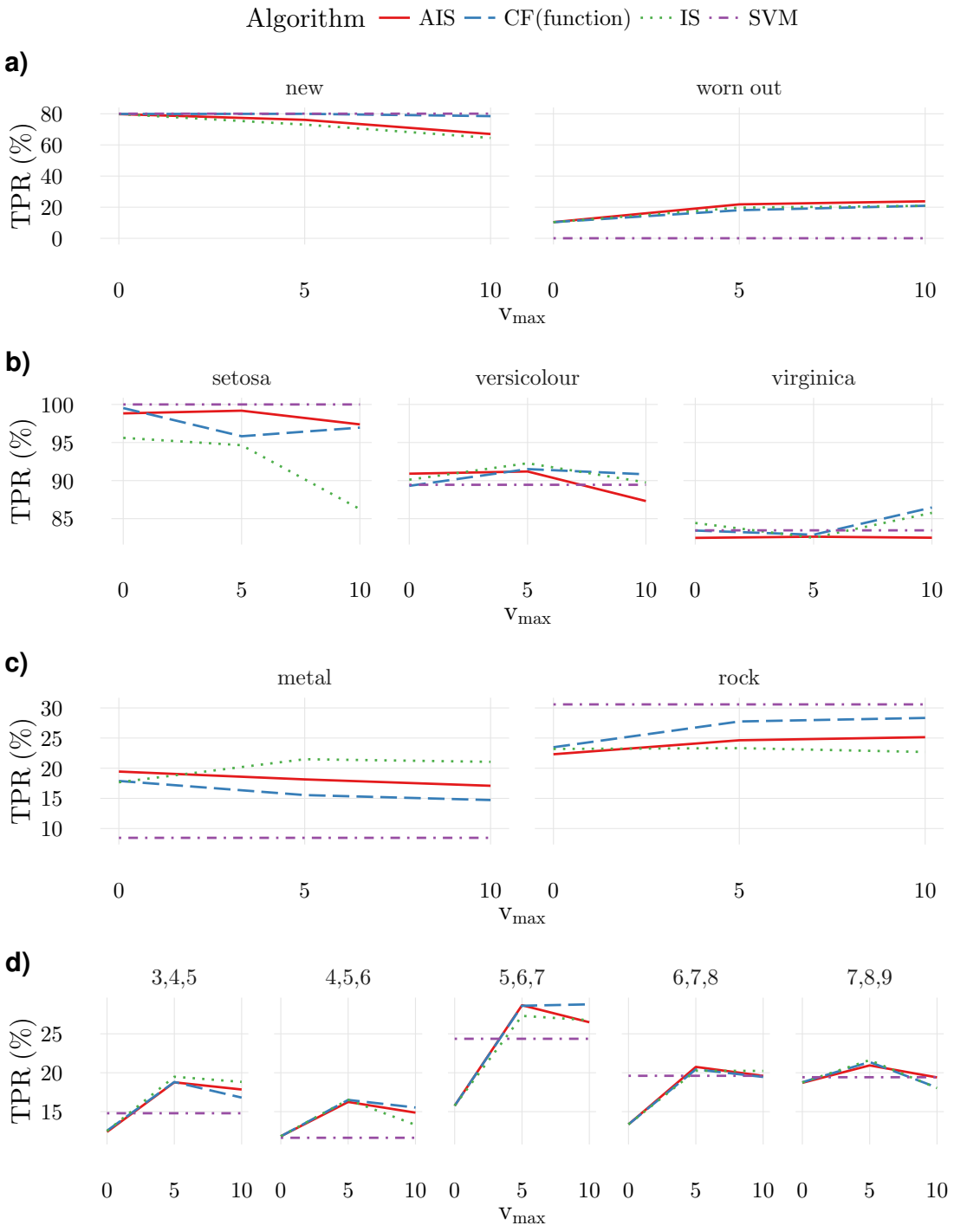


Figure A.1: Detection performance of the cellular frustration algorithm considering education with the immunological strategy (IS), artificial strategy (AIS), detection rule using expression (9.1) over the datasets considered for chapter 6 considering different values of $v_{max}$. All results considered a false positive rate of 10%. For reference it is also plotted the detection performance considering the one-class support vector machines. Overall, the detection rule achieved similar results to the CFA with agent dynamics.